

1. FONCTIONS

Solution L1

```
--> K = 0:1.5:2*pi
K =
    0.    1.5    3.    4.5    6.
--> L = linspace(0,2*pi,5)
L =
    0.    1.5707963    3.1415927    4.712389    6.2831853
```

Solution L2

```
--> X=10;
--> T0=1;
--> w0=2*pi/T0;

--> X*cos(w0*.5)
ans =
    - 10.

--> X*cos(w0*(linspace(0,2,20)))
ans =
    column 1 to 5
    10.    7.8914051    2.4548549    - 4.0169542    - 8.7947375
    column 6 to 10
    - 9.863613    - 6.7728157    - 0.8257935    5.4694816    9.4581724
    column 11 to 15
    9.4581724    5.4694816    - 0.8257935    - 6.7728157    - 9.863613
    column 16 to 20
    - 8.7947375    - 4.0169542    2.4548549    7.8914051    10.
```

Solution L3

```
--> V = [1;2;3]
V =
    1.
    2.
    3.

--> W = [1 2 3]'
W =
    1.
    2.
    3.
```

Solution F1

```
--> function x = x(t)
> x = X*cos(w0*t)
> endfunction

--> x(.5)
ans =
    - 10.

--> x(linspace(0,1,10)')
ans =
    10.
    7.6604444
    1.7364818
    - 5.
    - 9.3969262
    - 9.3969262
    - 5.
    1.7364818
    7.6604444
    10.
```

2. GRAPHES

Solution G1

abscisses :

```
t = linspace(0,6,15)'
```

ordonnées :

```
x(t)
```

Solution G2

Le script se termine par :

```
// intervalle de temps
t = linspace(0,6,1000)'; // ' est facultatif
// tracé
plot2d(t,x(t))
```

Solution G3

Le script se termine par :

```
t = linspace(0,6,1000)';
plot2d(t,x(t),2)
```

Solution G4

```
//graphes : options courantes

A = gca(); // accès aux propriétés des axes
A.x_location="origin";A.y_location="origin"; // les axes se coupent en 0
//A.isoview="on"; // base orthonormée

//on entre le titre souhaité dans la console :
titre = input("donner le titre du graphe (caractères ASCII) : ","s");
A.title.text=titre;
A.title.font_size=3; // taille du titre
titre_ok = input("texte du titre correct (o/n) ? ","s");
// on valide ou non le titre :
```

//tant que le titre n'est pas correct, on recommence

```
while titre_ok == "n"
    titre = input("donner le titre du graphe : ","s");
    A.title.text=titre;
    titre_ok = input("titre correct (o/n) ? ","s");
end
```

```
// position du titre
// à défaut de précision, position automatique = en haut et centré
postitre_ok = input("position correcte (o/n) ? ","s");
// on valide ou non la position du titre :
```

//tant que la position n'est pas correcte, on recommence

```
while postitre_ok == "n"
    x_titre = input("donner la position horizontale du titre : ");
    y_titre = input("donner la position verticale du titre : ");
    A.title.position=[x_titre,y_titre]; // position du titre
    postitre_ok = input("position correcte (o/n/a) [a = abandon, retour en position
automatique] ? ","s");
end
```

//si on abandonne, le titre sera placé automatiquement

```
if postitre_ok == "a"
    A.title.auto_position = "on"
end
```

```

// on étiquette les axes :
abcsisse = input("étiquette des abscisses (caractères ASCII) : ","s");
ordonnee = input("étiquette des ordonnées (caractères ASCII) : ","s");
A.x_label.text=abcsisse;
A.y_label.text=ordonnee;
A.x_label.font_size=3;A.y_label.font_size=3;

// autres options : trait, fond, boîte, grille
A.children.children.thickness=2; //épaisseur courbe(s)
//A.children.children.foreground=color(80,90,200); //couleur courbe(s)
A.background=color(249,248,230);
A.box="on";
A.thickness=1;A.foreground=color(90,90,90);
A.grid=[16,16];
A.grid_style=[7,7]; //propriété inconnue sur certaines versions de Scilab ?
A.grid_thickness=[1,1]; //propriété inconnue ?

```

3. EXPLOITATION PHYSIQUE

'oscillations_pseudopériodique - fonction.sce'

```

// Oscillation libres amorties - Régime pseudopériodique
// Définition

// position d'équilibre : xeq
// période propre : T
// facteur d'amortissement : ksi
// position initiale : x0
// vitesse initiale : v0
w = 2*pi/T // pulsation propre

function x = x(t)
    wp = w*sqrt(1-ksi^2); // pseudopulsation
    A = x0-xeq // coeff du cosinus
    B = (v0 + ksi*w*A)/w // coeff du sinus
    x = xeq + exp(-ksi*w*t) .* (A*cos(wp*t) + B*sin(wp*t))
endfunction

```

'oscillations_critique - fonction.sce'

```

// Oscillation libres amorties - Régime critique
// Définition

// position d'équilibre : xeq
// période propre : T
// facteur d'amortissement : ksi = 1
// position initiale : x0
// vitesse initiale : v0
w = 2*pi/T // pulsation propre

function x = x(t)
    A = v0+w*(x0-xeq) // coeff A
    B = x0-xeq // coeff B
    x = xeq + (A*t + B) .* exp(-w*t)
endfunction

```

'oscillations_apériodique - fonction.sce'

```

// Oscillation libres amorties - Régime apériodique
// Définition

// position d'équilibre : xeq
// période propre : T
// facteur d'amortissement : ksi
// position initiale : x0

```

```

// vitesse initiale : v0
w = 2*pi/T // pulsation propre

function x = x(t)
d1 = w*(ksi-sqrt(ksi^2 -1)) // delta 1
d2 = w*(ksi+sqrt(ksi^2 -1)) // delta 2
A = (d2*(x0-xeq)+v0)/(d2-d1) // coeff A1 de la 1e exp
B = (d1*(x0-xeq)+v0)/(d1-d2) // coeff A2 de la 2e exp
x = xeq + A*exp(-d1*t) + B*exp(-d2*t)
endfunction

```

Oscillations libres amorties : illustration graphique de l'influence du facteur d'amortissement (avec 3 fonctions)

```

//oscillations libres amorties
//graphes pour ksi=0,1 ; 0,5 ; 1 et 2

mode(0) // Affichage des résultats dans la console
clear // Réinitialisation de toutes les variables
xdel(winsid()) // Fermeture de toutes les figures
funcprot(0) // Désactivation de la protection des fonctions

// paramètres (tester d'autres valeurs)
xeq = 0;
T = 1;
x0 = 1;
v0 = 0;

// appel fonction - régime aperiodique
ksi = 2;
exec('...chemin\oscillations_régime aperiodique.sce', -1) //chemin à corriger
// intervalle de temps
t = linspace(0,2,1000)';
Aper=x(t);

// appel fonction - régime critique
exec('...chemin\oscillations_régime critique.sce', -1) //chemin à corriger
Crit=x(t);

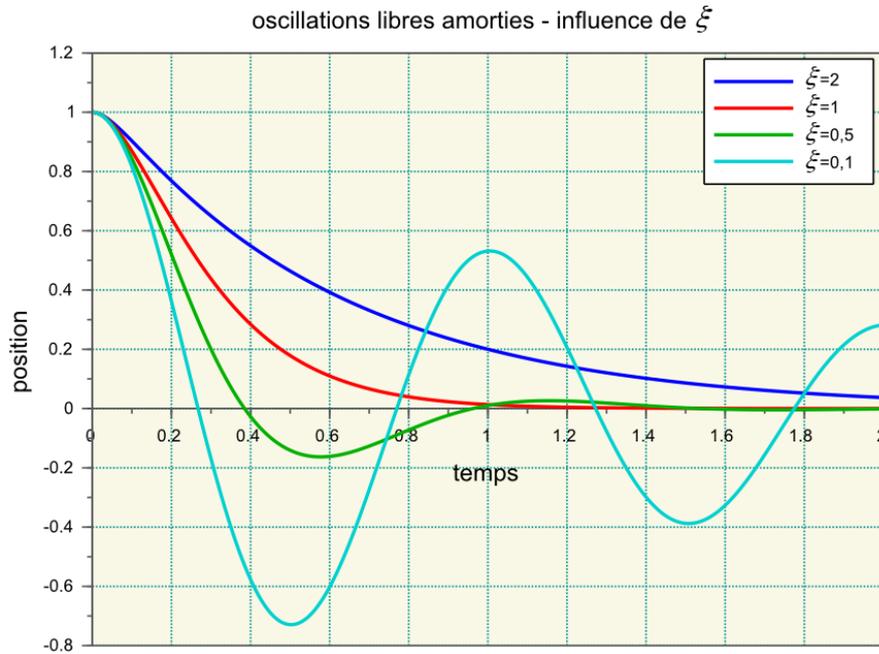
// appel fonction - régime pseudoperiodique
ksi = 0.5; //1er cas
exec('...chemin\oscillations_régime pseudoperiodique.sce', -1) //chemin à corriger
Pseudoper=x(t);
ksi = 0.1; //2e cas
Pseudoper2=x(t);

// tracé
plot2d(t, [Aper,Crit,Pseudoper,Pseudoper2], [2,5,14,18])
hl=legend(["ksi=2";"ksi=1";"ksi=0,5";"ksi=0,1"]);

//exec('...chemin\options_graphes.sce', -1) //chemin à corriger

funcprot(1) // Réactivation de la protection des fonctions

```



Oscillations libres amorties : illustration graphique de l'influence du facteur d'amortissement (avec 1 seule fonction)

```
//oscillations libres amorties
//illustration graphique de l'influence du facteur d'amortissement

mode(0)          // Affichage des résultats dans la console
clear           // Réinitialisation de toutes les variables
xdel(winsid())  // Fermeture de toutes les figures

// paramètres (tester d'autres valeurs)
xeq = 0.1;      // position d'équilibre
T = 0.628;     // période propre
x0 = 0.5;      // position initiale
v0 = 10;       // vitesse initiale

// définition de la fonction
// ici on copie la fonction

// intervalle de temps
t = linspace(0,2,1000)';

// régime aperiodique
ksi = 2;       // facteur d'amortissement
Aper=x(t);

// régime critique
ksi = 1;
Crit=x(t);

// régime pseudoperiodique
ksi = 0.5;    //1er cas
Pseudoper=x(t);
ksi = 0.1;    //2e cas
Pseudoper2=x(t);

// tracé
plot2d(t, [Aper,Crit,Pseudoper,Pseudoper2], [2,5,14,18])
hl=legend(["ksi=2";"ksi=1";"ksi=0,5";"ksi=0,1"]);

//exec('...chemin\options_graphes.sce', -1) //chemin à corriger
```