

**Contexte :**

L'affichage de graphique et de dessins sous Scilab offre de très nombreuses possibilités. Les exercices qui suivent donnent quelques exemples.

Pour plus d'information, tapez : **help plot** dans la console Scilab et allez sur le site : <http://www.juggling.ch/gisin/coursam4os/index.html> info sur **plot** qui suit la série 3.

Il est conseillé de faire les exercices suivants dans un ou plusieurs fichiers SciNotes

**1. Un premier graphique simple**

Recopier les trois lignes suivantes pour les exécuter ensuite :

```
ax1 = linspace(0, 7, 101); // 101 points régulièrement espacés entre 0 et 7
ay1 = sin(ax1);
plot(ax1, ay1);
```

Remarque :

Pour comprendre ce que fait l'instruction `ax1 = linspace(0, 7, 101);` tapez les lignes

```
ax1b = 0:0.07:7;
disp(ax1b - ax1);
```

On remarque que les vecteurs `ax1` et `ax2` sont presque identiques. Ils contiennent 101 nombres régulièrement espacés entre 0 et 7.

**2. Un graphique avec contrôle de couleur, épaisseur et style**

Recopier les lignes suivantes pour les exécuter ensuite :

```
scf(2); // Dessinera dans la fenêtre graphique n° 2.
clf(2); // Efface le contenu de la fenêtre n° 2.
ax2 = linspace(-20, 20, 400); // points régulièrement espacés
ay2 = sin(ax2) ./ ax2;

plot( ax2, ay2..
    , "color", [0, 0, 1].. // couleurs [r, g, b] entre 0 et 1 ou une lettre "r", "g", "b".
    , "Thickness", 4.. // épaisseur du trait
    , "LineStyle", ":".. // "-" ligne pleine, "--" traitillé, ":" pointillé, "-.", "none"
    , "Visible", "on".. // si on veut cacher une ligne, mettre "off"
    );
```

Après avoir testé les instructions ci-dessus, ajoutez celles ci-dessous.

```
ax2b = linspace(-6*pi, 6*pi+1e-8, 25); // points régulièrement espacés
ay2b = sin(ax2b) ./ ax2b;
plot( ax2b, ay2b, 'LineStyle', 'none'..
    , "Marker", 'o'.. // "o", "x", "*", ".", "d"=diamond, "s"=square
    , "MarkerSize", 8 ..
    , "MarkerFaceColor", 'r'.. // "r", "g", "b", "c", "m", "k"=black, "w"=white
    , "MarkBackground", "m".. // fond de la marque, = MarkerFaceColor
    , "MarkForeground", "k".. // bord
    );
```

Dans la console Scilab, tapez : **help LineStyle** pour tous les types de lignes possibles. De l'aide sur d'autres paramètres est disponible au même endroit.

Remarquez que si vous fermez la fenêtre graphique n°0, mais pas la n°2, en exécutant l'intégralité du code de SciNotes, vous n'avez plus de graphique n°0. Il est donc conseillé d'ajouter les instructions :

```
scf(1); // Dessinera dans la fenêtre graphique n° 1.
clf(1); // Efface le contenu de la fenêtre n° 1.
au début des instructions de l'exercice 1.
```

### 3. Plusieurs courbes dans un graphique, avec des légendes.

Recopier les lignes suivantes pour les exécuter ensuite :

```
// Plusieurs courbes dans un graphique, avec des couleurs et des légendes
win3 = scf(3); // Mémoire un pointeur sur la fenêtre n° 3
clf(3); // Efface le contenu de la fenêtre n° 3.
ax3a = linspace(-2, 2, 101); // points régulièrement espacés
ay3a = exp(ax3a); // pour la courbe de l'exponentielle
plot( ax3a, ay3a..
      , "color", [0.5, 0, 0].. // couleurs [r, g, b].
      , "Thickness", 1.. // épaisseur du trait
      , "LineStyle", "-".. // "-" ligne pleine, "--" traitillé, ":" pointillé, "-.", "none"
    );

// Inversion de X et Y, donne la fonction Ln
plot( ay3a, ax3a..
      , "color", [0, 0.5, 0]..
      , "Thickness", 1..
      , "LineStyle", "-"..
    );

// Ligne de symétrie y = x
plot( [-2, 8], [-2, 8]..
      , "color", [0, 0.5, 0.5]..
      , "Thickness", 1..
      , "LineStyle", "--"..
    );

// Couleur du fond de la fenêtre 'win3'
win3.color_map(31,:) = [1, 1, 0]; // = jaune
win3.background = 31; // référence à la table de couleur 'color_map'

// Référence au graphique
axes3 = win3.children; // la courbe et les axes contenu dans la fenêtre graphique

// Couleur du fond du graphique
win3.color_map(30,:) = [0.5, 1, 1]; // = cyan clair
axes3.background = 30; // change la couleur de fond du graphique.

axes3.thickness = 2; // épaisseur des axes
axes3.font_size = 3; // Taille de la police de caractères
axes3.font_style = 4;
// 0=courrier gras, 1=courrier gras, 2=courrier, 3=italique
// 5=gras+italique, 4=gras, 6=normal, 7=gras italique
// 8=très gras, 9=très gras + italique
// c.f. help font_style et help graphics fonts

axes3.zoom_box=[-2.1, -2.1, 7.6, 7.6]; // défini les échelles des axes

// Titre, légendes et informations.
title( axes3, "C'est le titre", ..
      'fontsize', 5);
xlabel(axes3, "axe des x", 'fontsize', 4);
ylabel(axes3, "axe des y", 'fontsize', 4);
xstring(-0.2, 6, "y = exp(x)");
xstring(5, 5, "y = x");
xstring(5.3, 1, "y = Ln(x)");

win3.figure_size = [780, 520]; // c.f. help figure_size
axes3.margins = [0.08, 0.02, 0.1, 0.15]; // gauche, droite, haut, bas
```

Une fois que votre code fonctionne, faites des essais, modifiez les couleurs de fond, les légendes, les positions des textes, leur taille, leur police, etc.

Dans la console Scilab, tapez win3 pour de l'information. Idem pour axes3.

#### 4. Graphisme 3D

Voici un exemple de code permettant de tracer une fonction en 3D

```
win4 = scf(4); // Mémoire un pointeur sur la fenêtre n° 4
clf(4); // Efface le contenu de la fenêtre n° 4.
ax4 = linspace(-4, 4, 21);
ay4 = linspace(-6, 6, 31);
az4 = cos(ax4') * cos(ay4); // = Matrice correspondant à la fonction de x et y
plot3d(ax4, ay4, az4);

axes4 = win4.children; // pour modifier des données sur le graphique
data4 = axes4.children;

// Change la couleur de la surface
win4.color_map(31,:) = [0, 1, 0]; // = verte
data4.color_mode = 31;

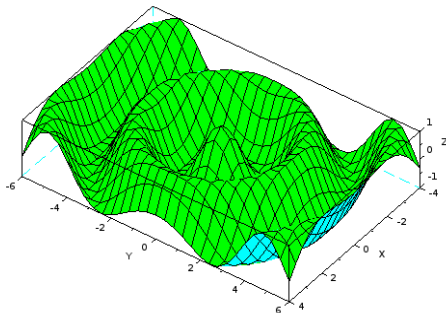
// Change la couleur du dessous de la surface
win4.color_map(32,:) = [1, 0, 1]; // = magenta
data4.hiddencolor = 32;

data4.thickness = 2; // épaisseur des traits
```

Pour des informations sur divers structures, dans la console SciLab, tapez :

win4 ou axes4 ou data4

**5. Challenge** : Sauriez-vous tracer le graphique suivant, correspondant à :  $f(x, y) = \cos(2 \cdot \sqrt{x^2 + y^2})$



## 6. Animation graphique

Voici un exemple de code permettant de faire une animation

```
win6 = scf(6); // Mémoire un pointeur sur la fenêtre n° 6
clf(6); // Efface le contenu de la fenêtre n° 6.
at6 = linspace(0, 30, 201)';
ax6 = ( at6 / 10 ) .* cos(at6);
ay6 = ( at6 / 10 ) .* sin(at6);

plot( ax6, ay6);

axes6 = win6.children; // pour modifier des données sur le graphique
data6 = axes6.children;
line6 = data6.children; // les données qui sont dans le graphique

data6.children.thickness = 5; // épaisseur du trait

// modifie les données de la courbe pour faire une animation
for nn=1:50 do
    drawlater(); // pour différer l'affichage graphique
    line6.data = [( at6 / 10 ) .* cos(at6 - 0.05*nn) , ( at6 / 10 ) .* sin(at6 - 0.05*nn)];
    drawnow(); // pour afficher le graphique avec les modifications de la courbe
end;
```

## 7. Animation graphique plus simple

En vous basant sur le code de l'exercice 6, pouvez vous animer la fonction :

$$f_t(x) = \sin(x + t), \quad x = 0:0.1:20;$$

Le "temps"  $t$  peut sauter de 0.1 à chaque itération pour animer la fonction, qui représentera un sinus qui se déplace.

Le code est similaire et un peu plus simple que celui de l'exercice 6.