

## FEUILLE D'EXERCICES n° 10

### Représentations graphiques

Dans l'aide (`plot?`), on trouve des explications et des exemples d'utilisation des commandes graphiques `plot`, `line`, `point`, `text`, etc. Voyons ici quelques exemples.

1) Exécuter les commandes suivantes.

```
plot(sin)
plot(sin, (0,10))
plot(sin(1/x))
```

La fonction `plot` semble s'appliquer à l'intervalle  $[-1, 1]$  par défaut. On peut aussi bien utiliser `sin` ou `sin(x)` dans `plot`.

2) Pour dessiner un nuage de points :

```
point((-1,-1), (7,0), (1,5.3), (6,8))
```

ou bien, si l'on n'y voit pas très bien :

```
point((-1,-1), (7,0), (1,5.3), (6,8), color='red', pointsize=20)
```

On peut aussi donner un nom à cette figure

```
p = point((-1,-1), (7,0), (1,5.3), (6,8), color='red', pointsize=20)
```

3) Pour dessiner des segments, on dispose de la commande `line`.

```
l = line((-1,-1), (7,0), (1,5.3), (6,8)); l
```

On peut alors dessiner `p` et `l` sur la même figure.

```
lp = p+l; lp
```

On peut aussi ajouter du texte.

```
t = text('A', (-1.2,-1), color='red')
t += text('B', (7,0.35), color='red')
t += text('C', (0.8,5.3), color='red')
t += text('D', (6.2,8), color='red')
lp + t
```

4) Pour un exemple plus joli, exécuter les commandes suivantes (prises dans l'aide).

```
p = plot(x^2, (-0.5, 1.4)) + line([(0,0), (1,1)], color='green')
p += line([(0.5,0.5), (0.5,0.5^2)], color='purple')
p += point([(0,0), (0.5,0.5), (0.5,0.5^2), (1,1)], color='red', pointsize=20)
p += text('A', (-0.05, 0.1), color='red')
p += text('B', (1.01, 1.1), color='red')
p += text('C', (0.48, 0.57), color='red')
p += text('D', (0.53, 0.18), color='red')
p
p.show(axes=False, xmin=-0.5, xmax=1.4, ymin=0, ymax=2)
```

5) Il y a différentes manières d'indiquer les bornes d'un graphe. Dans l'aide, on trouve les commandes suivantes.

```
plot(lambda x: x, (x,-1,1))
plot(lambda x: x, -1,1)
plot(x,x,-1,1)
plot(x,-1,1)
plot(x^2, x, -1,1)
plot(x^2, xmin=-1,xmax=2)
```

```

plot(lambda x: x, x, -1,1)
plot(lambda x: x^2, x, -1,1)
plot(lambda x: 1/x, x, -1,1)
f(x) = sin(x+3) - .1*x^3
plot(lambda x: f(x), x, -1,1)

```

6) On peut aussi dessiner deux courbes sur un même graphe de la manière suivante.

```

plot([sin(x), cos(2*x)*sin(4*x)], -pi, pi)

```

7) Comment dessiner une courbe paramétrée? Avec `parametric_plot`. Voici donc une autre façon de dessiner un cercle :

```

parametric_plot((cos(t), sin(t)), (t, 0, 2*pi))

```

Un exemple plus amusant :

```

parametric_plot([cos(x) + 2*cos(x/4), sin(x) - 2*sin(x/4)], (x,0, 8*pi))

```

Un exemple où l'on recolle deux branches :

```

P1 = parametric_plot((3*t/(t^3+1), 3*t^2/(t^3+1)), (t,-0.8,10))
P2 = parametric_plot((3*t/(t^3+1), 3*t^2/(t^3+1)), (t,-10,-1.2))
P1 + P2

```

8) Soit  $\mathcal{C}$  la courbe d'équation paramétrée

$$\begin{cases} x = 4t^3 \\ y = 3t^4 \end{cases}$$

Dessiner la courbe  $\mathcal{C}$ .

9) Restent encore les courbes paramétrées en coordonnées polaires : `polar_plot`.

```

polar_plot(sin(5*x)^2, (x, 0,2*pi), color='red')

```

Dessiner la courbe d'équation polaire

$$r = \frac{\cos(2\theta)}{\cos \theta}$$

10) Une autre commande intéressante : `implicit_plot`. Par exemple, essayer

```

var("x y")
implicit_plot((x^2+y^2-2, (x,-3,3), (y,-3,3)))

```

On peut également définir la fonction préalablement :

```

f(x,y) = x^2 + y^2 - 2
implicit_plot((f(x,y), (x,-3,3), (y,-3,3)),fill=true)

```

Exemple d'une figure composée de plusieurs graphes (issue de l'aide)

```

G = Graphics()
counter = 0
for col in colors.keys():
    G += implicit_plot(x^2+y^2==1+counter*.1, (x,-4,4), (y,-4,4),color=col)
    counter += 1
G.show(frame=False)

```

Ici, `Graphics` désigne le graphe vide. Il sert à donner une première valeur à `G`, avant de le faire varier dans la boucle.

11) Dessiner sur un graphe un triangle et son cercle circonscrit.

12) Dessiner la courbe d'équation

$$x^3 + y^3 - 3 * x * y.$$