

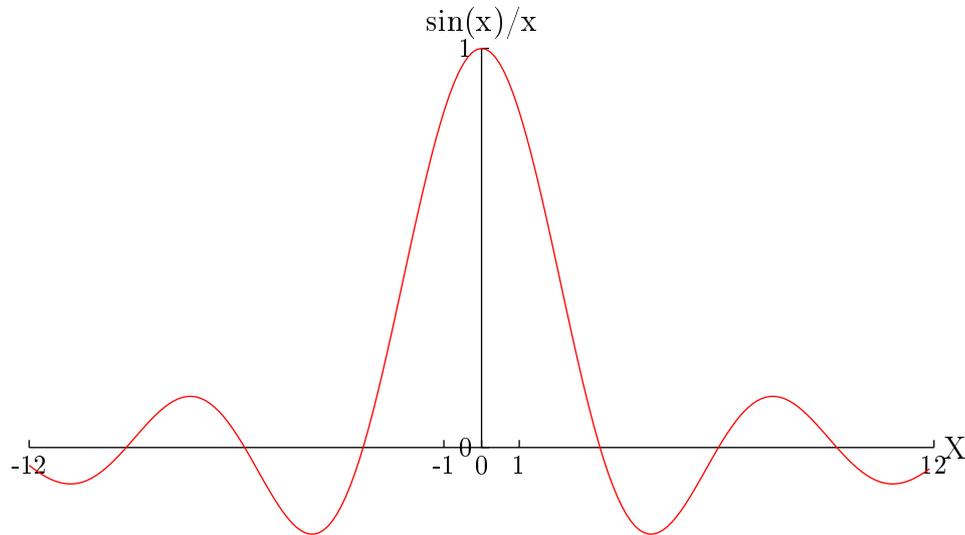
## Desin vectoriel : tracé de courbes

### 1. Tracé de courbes avec Python

Pour tracer une courbe avec Python et le module `draw`, il faut tout d'abord définir la transformation affine avec l'élément `plotrange`. Un élément `polyline` peut être généré par un code Python puis inséré à l'emplacement adéquat par un élément `python` de type `xml`. L'exemple suivant trace la fonction  $(\sin(x)/x)$  entre  $-4\pi$  et  $4\pi$ .

```
<python visible="false">import math
n=200
points=""
for k in range(n):
    x = -12.0 + k*24.0/n
    if x!=0:
        y = math.sin(x)/x
    else:
        y = 1
    points += "%f,%f"%(x,y)
polyline = "<polyline points='%s'/">%points</python>
<draw name="figureA" width="800" height="600" medium-font-size="20" pdf-scale="1" str
  <translate tx="100" ty="100">
    <plotrange xmin="-12" xmax="12" ymin="-0.5" ymax="1" width="600" height="400"
      <haxis x="-12,0,12" y="0" fill="black" font-size="x-small">
        <legend pos="end" font-size="medium">X</legend>
      </haxis>
      <vaxis x="0" y="0,1" fill="black" font-size="x-small">
        <legend pos="end" font-size="medium">sin(x)/x</legend>
      </vaxis>
      <g fill="none" stroke="red">
        <python type="xml">polyline</python>
      </g>
    </plotrange>
  </translate>
</draw>
```

```
import math
n=200
points=""
for k in range(n):
    x = -12.0 + k*24.0/n
    if x!=0:
        y = math.sin(x)/x
    else:
        y = 1
    points += "%f,%f"%(x,y)
polyline = "<polyline points='%s'/">%points
```



On remarquera l'absence d'espace juste après la balise ouvrante de l'élément `python` et juste avant la balise fermante. Bien sûr, le code Python doit être indenté correctement.

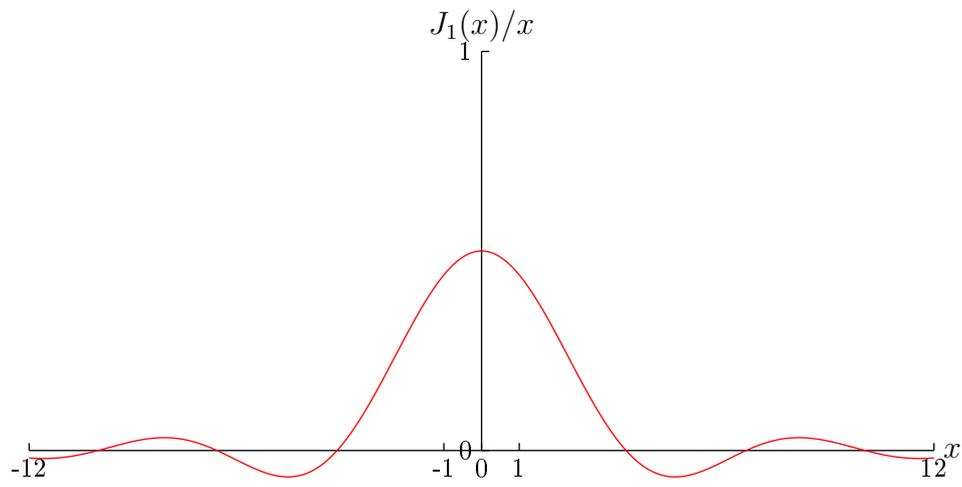
## 2. Tracé de courbes avec Scilab

La technique précédente peut être utilisée avec Scilab. Pour cela, il faut calculer les points à tracer puis générer une chaîne de caractères définissant un élément `polyline` que l'on insère au bon endroit avec l'élément `scilab` muni de l'attribut `type='xml'`.

L'exemple suivant effectue le tracé de la fonction  $J_1(x)/x$ , où  $J_1$  est une fonction de Bessel de premier type (fonction non disponible en Python standard).

Voici tout d'abord le code scilab générant l'attribut `points` et l'élément `polyline` :

```
n=200
x = linspace(-12,12,n);
points = ''
for k=1:n,
    points = points + string(x(k))+','+string(besselj(1,x(k))/x(k))+' ';
end
polyline = '<polyline '+'points='''+points+'''/>';
```



Une autre solution, plus simple mais moins souple, consiste à exporter un tableau puis à tracer la courbe avec un élément `plot2d` (exemple [ici](#))