Bibliothèques TP7

Exercice 1 Premières fonctions non natives

Q1 Importer la bibliothèque `math`. Elle contient la fonction `exp` et la fonction `sin`. Écrire une fonction Python `f` prenant en argument un réel x et renvoyant $\exp(\sin^2(x))$. On vérifiera que `f(12)` vaut approximativement 1,333.

```
import math
def f(x):
    return math.exp(math.sin(x)**2)
```

Q2 Importer la bibliothèque `numpy` avec l'alias `np`. Cette bibliothèque contient la constante `pi` et la fonction racine carrée `sqrt` (pour l'anglais “square root”). Écrire une commande permettant d'accéder à la valeur de $\sqrt{\pi}$. On vérifiera que cette valeur vaut environ 1,772. Accéder ensuite à la valeur de $\ln(\pi) \simeq 1,1447$.
À retenir : en Python $\ln(x)$ s'écrit : `np.log(x)`

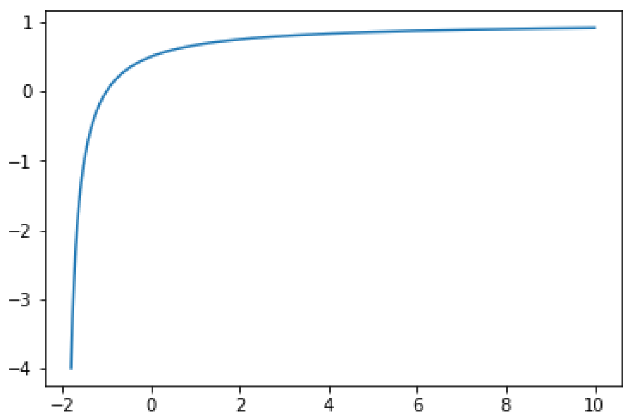
```
import numpy as np
print(np.sqrt(np.pi))
print(np.log(np.pi))
```

Exercice 2 Un premier graphique

Q1 Prédire ce que contient `np.linspace(0,1,5)`. Contient 0, 0.25, 0.5, 0.75 et 1 (5 valeurs en tout).

Q2 Tracer le graphe de la fonction $g : x \longmapsto (x+1)/(x+2)$ sur $[-1.8, 10]$ en utilisant 1000 points.

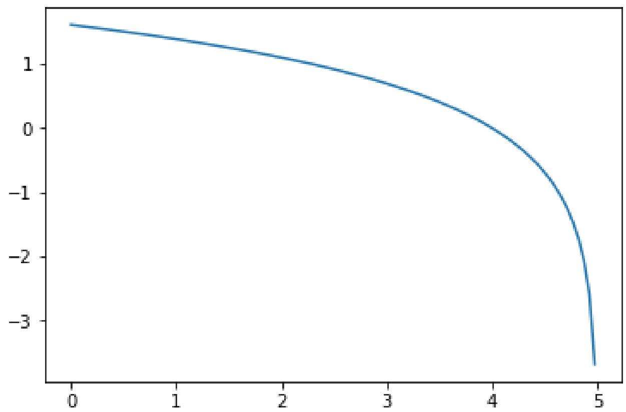
```
def g(x):
    return (x+1)/(x+2)
absi = np.linspace(-1.8,10,1000)
ordo = g(absi)
plt.plot(absi, ordo)
plt.show()
```



Q3 Tracer le graphe de la fonction $h : x \longmapsto \ln(5-x)$ sur $[0,10]$ avec 200 points de tracé. Votre graphe est-il bien sur l'intervalle $[0,10]$? Qu'a fait automatiquement Python ?

```
def h(x):
    return np.log(x-3)
absi = np.linspace(0,100,200)
ordo = h(absi)
plt.plot(absi,ordo)
plt.show()
```

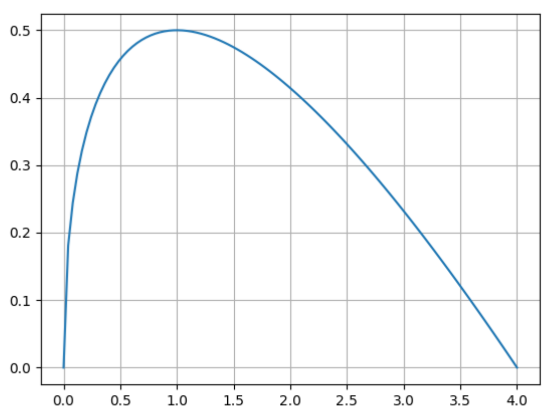
Python n'a tracé que là où $h(x)$ avait un sens, sans même nous dire de message d'erreur !



Exercice 3 Fonctions usuelles, symétries, translations

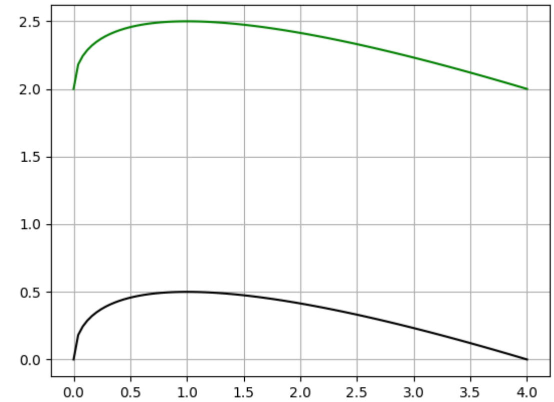
Q1 Définir une fonction Python `f` correspondant à la fonction $f : x \longmapsto \sqrt{x} - \frac{x}{2}$. Tracer ensuite le graphe de f sur $[0,4]$ avec 100 points de tracé et en affichant une grille.

```
# Ne pas oublier de réimporter les bibliothèques !
import numpy as np
import matplotlib.pyplot as plt
def f(x):
    return np.sqrt(x)-x/2
absi = np.linspace(0,4,100)
ordo = f(absi)
plt.plot(absi,ordo)
plt.grid()
plt.show()
```

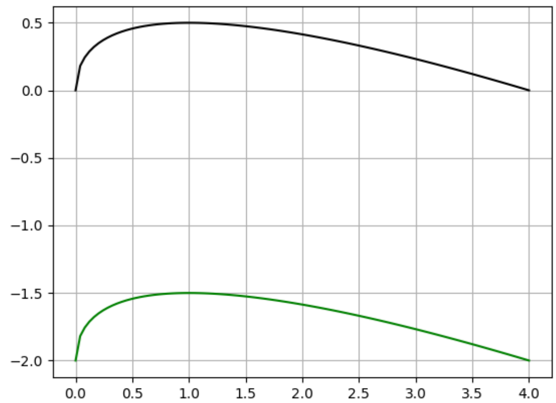


Q2 Définir une fonction Python correspondant à la fonction $g : x \longmapsto f(x) + 2$. Tracer alors <u>sur le même dessin</u> les graphes de f et de g sur $[0, 4]$ avec 100 points de tracé. On tracera le graphe de f en noir et celui de g en vert, et on affichera une grille.

```python
def g(x):
    return f(x)+2
absi = np.linspace(0,4,100)
ordo1 = f(absi)
ordo2 = g(absi)
plt.plot(absi,ordo1,'k')
plt.plot(absi,ordo2,'g')
plt.grid()
plt.show()
```



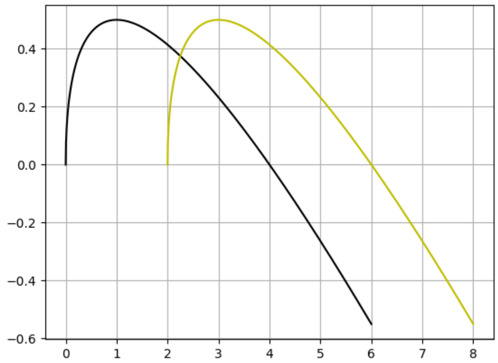
Q3 Reprendre la question précédente mais avec g définie par $g : x \longmapsto f(x) - 2$. De manière générale, comment obtient-on le graphe de $x \longmapsto f(x) + K$ en fonction de celui de f ?



Le graphe de $x \longmapsto f(K) + K$ s'obtient par translation vericale de K.

Q4 Définir une fonction Python correspondant à la fonction $h : x \longmapsto f(x - 2)$. Tracer alors <u>sur le même dessin</u> les graphes de f et de h sur des ensembles appropriés avec 1000 points de tracé. On tracera le graphe de f en noir et celui de h en jaune, et on affichera une grille. Reprendre cette question avec $h : x \longmapsto f(x + 2)$ en modifiant si besoin les ensembles de tracés pour visualiser correctement le dessin. De manière générale, comment obtient-on le graphe de $x \longmapsto f(x + K)$ en fonction de celui de f ?

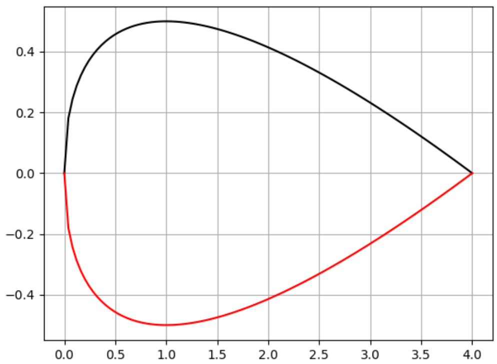
```
def h(x):
    return f(x-2)
absi1 = np.linspace(0,6,1000)
absi2 = np.linspace(2,8,1000)
ordo1 = f(absi1)
ordo2 = h(absi2)
plt.plot(absi1,ordo1,'k')
plt.plot(absi2,ordo2,'y')
plt.grid()
plt.show()
```

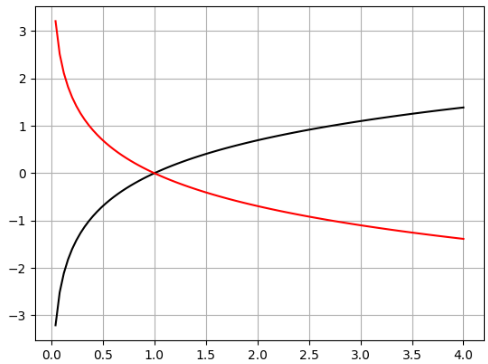


Le graphe de $x \longmapsto f(x+K)$ s'obtient par translation horizontale du graphe de f de $-K$.

Q5 Définir une fonction Python correspondant à la fonction $f_\ominus : x \longmapsto -f(x)$. Tracer alors sur le même dessin les graphes de f et de $f_\ominus$ sur des ensembles appropriés. On tracera le graphe de f en noir et celui de $f_\ominus$ en rouge, et on affichera une grille. Reprendre cette question avec la fonction $f^{\text{bis}} : x \longmapsto \ln(x)$. De manière générale, comment obtient-on le graphe de $x \longmapsto -f(x)$ en fonction de celui de f ?

```
def fmoins(x):
    return -f(x)
absi = np.linspace(0,4,100)
ordo = fmoins(absi)
ordo2 = f(absi)
plt.plot(absi,ordo2,'k')
plt.plot(absi,ordo,'r')
plt.grid()
plt.show()
```

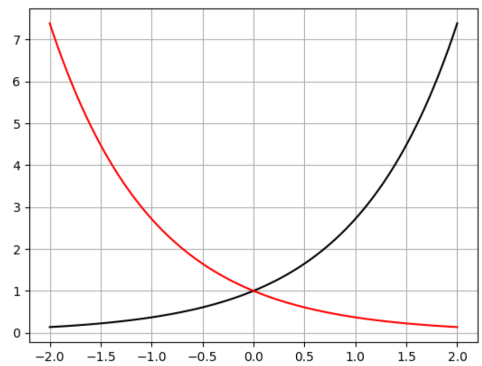


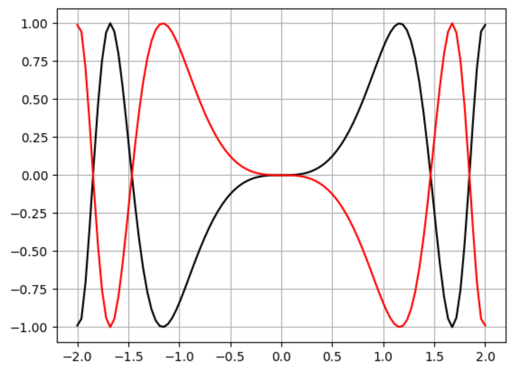


Le graphe de $-f$ s'obtient depuis celui de f par symétrie par rapport à l'axe des abscisses.

Q6 Dans cette question, on utilise la fonction $p : x \longmapsto e^x$. Définir la fonction p en Python, ainsi qu'une fonction correspondant à $\tilde{p} : x \longmapsto p(-x)$. Tracer alors sur le même dessin les graphes de p et de $\tilde{p}$ sur $[-2, 2]$. On tracera le graphe de p en noir et celui de $\tilde{p}$ en rouge, et on affichera une grille. Reprendre cette question avec la fonction $p^{\text{bis}} : x \longmapsto sin(x^3)$. De manière générale, comment obtient-on le graphe de $x \longmapsto f(-x)$ en fonction de celui de f ?

```
def p(x):
    return np.exp(x)
def ptilde(x):
    return p(-x)
absi = np.linspace(-2,2,100)
ordo = p(absi)
ordo2 = ptilde(absi)
plt.plot(absi,ordo,'k')
plt.plot(absi,ordo2,'r')
plt.grid()
plt.show()
```





Le graphe de $\tilde{f}$ s'obtient depuis celui de f par symétrie par rapport à l'axe des ordonnées.

Q7 Que se passe-t-il lorsque f est paire ? impaire ?
Lorsque f est paire, le graphe de f et celui de $\tilde{f}$ sont égaux, donc ce graphe est symétrique par rapport à l'axe des ordonnées. Et lorsque f est impaire, c'est le graphe de f et celui de $-\tilde{f}$ qui sont égaux, donc le graphe de f reste le même après une symétrie par rapport à l'axe des abscisses suivie d'une symétrie par rapport à l'axe des ordonnées, c'est-à-dire après une symétrie centrale !

Q8 En appliquant les règles ci-dessus, tracez à la main les graphes de fonctions suivantes. Vérifiez ensuite votre résultat en utilisant Python.

1. $f_1 : x \longmapsto x^3 + 1$
2. $f_2 : x \longmapsto \ln(x + 1)$
3. $f_3 : x \longmapsto e^{-x}$
4. $f_4 : x \longmapsto -\dfrac{1}{x - 2}$

1. Le graphe de f_1 s'obtient par translation verticale de 1 par rapport au graphe de la fonction cube.
2. Le graphe de f_2 s'obtient par translation horizontale de -1 par rapport au graphe de la fonction logarithme.
3. Le graphe de f_3 s'obtient par symétrie par rapport à l'axe des ordonnées par rapport au graphe de l'exponentielle.

4. Pour obtenir le graphe de f_4 on : trace le graphe de la fonction inverse, on le décale de 2 vers la droite pour avoir le graphe de $x \longmapsto \dfrac{1}{x-2}$, puis on prend le symétrique de ce graphe par rapport à l'axe des abscisses pour avoir celui de $f_4 : x \longmapsto -\dfrac{1}{x-2}$.

```
def f1(x):
    return x**3+1
def f2(x):
    return np.log(x+1)
def f3(x):
    return np.exp(-x)
def f4(x):
    return -1/(x-2)
x = np.linspace(-2,2,100) # à ajuster pour chaque fonction
y = f1(x) # respectivement y = fk(x) pour chaque k
plt.plot(x,y)
plt.show()
```

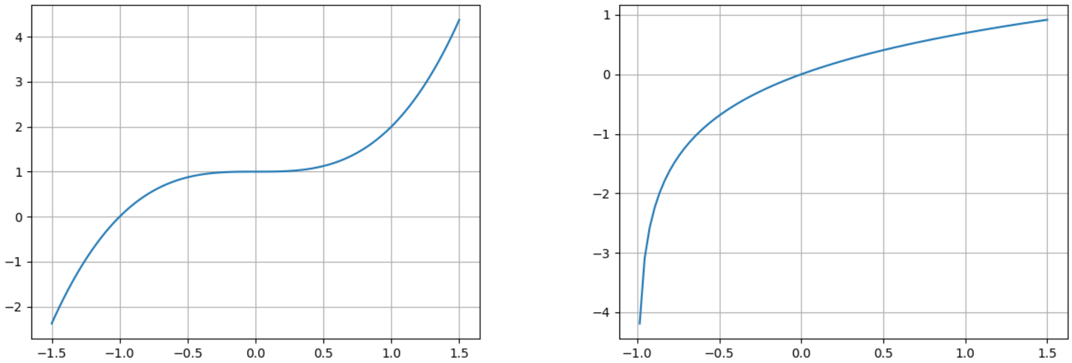


Figure 1 – f_1 (à gauche) et f_2 (à droite)

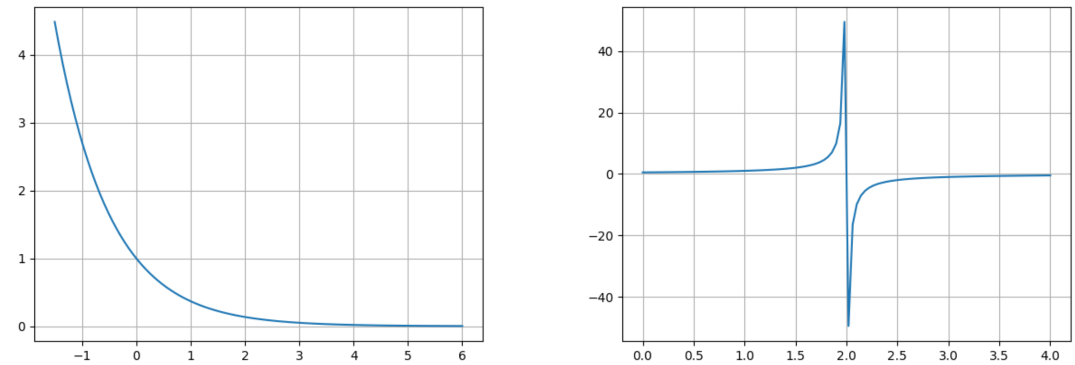


Figure 2 – f_3 (à gauche) et f_4 (à droite)