

TP 2 – Fonctions : images et antécédents

Pour ce TP, il sera nécessaire de charger les packages suivants.

```
1 import matplotlib.pyplot as plt
2 import numpy as np
```

Dans tout le TP, les fonctions f et g désignent les fonctions définies sur \mathbb{R} par :

$$f(x) = 2x + 1 \quad \text{et} \quad g(x) = \frac{x^2 + x + 1}{x^2 + 1}$$

1 Déterminer une image

1. Déterminer l'image de 2 par f .
2. L'algorithme suivant permet de créer une procédure, nommée f , qui demande un nombre réel x à l'utilisateur et qui renvoie l'image de x par f .

```
3 def f(x):
4     return (2*x+1)
```

Recopier l'algorithme et le tester pour $x = 2$.

3. Modifier l'algorithme précédent pour calculer l'image de 4 par la fonction g .

2 Tracer la courbe d'une fonction

L'algorithme suivant permet de tracer la courbe représentative de f à l'aide de Python.

```
5 x=np.linspace(-2,2,20)
6 plt.plot(x,f(x),'.')
7 plt.show()
```

1. Expliquer ce que signifie chaque ligne de l'algorithme.
2. Recopier l'algorithme et tracer la courbe de f .
Modifier enfin l'algorithme afin qu'il trace la courbe représentative de g .

3 Déterminer des antécédents

1. Déterminer par le calcul les éventuels antécédents de 3 par la fonction f .
2. L'algorithme suivant permet de créer une procédure, nommée `antecedent`, qui demande un nombre réel y à l'utilisateur et qui renvoie une valeur approchée (à 0,1 près) d'un antécédent de y par f .

```
5 def antecedent(y):  
6     a=0  
7     while f(a)>0:  
8         a=a-0.1  
9     return(a)
```

Recopier l'algorithme sur Python et le tester pour $y = 3$.

Remarque. Cette procédure ne permet pas de connaître le nombre d'antécédent mais seulement d'avoir une valeur approchée d'un des antécédents lorsque cela est possible.

3. Expliquer chaque ligne de l'algorithme puis expliquer pourquoi il calcule bien un antécédent de y par f .
4. Modifier l'algorithme précédent pour calculer une valeur approchée d'un antécédent de 0,8 par la fonction g . Déterminer une valeur approchée à 0,1 près puis à 0,01 près.