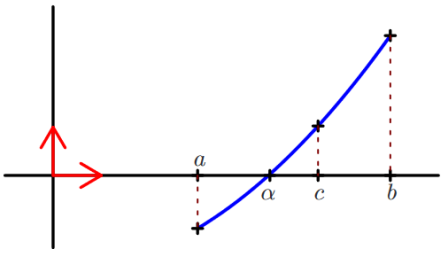
Exercice 1

1. Que fait ce programme pour différentes valeurs entières de n et d ?

```
n,d = 8, 3
pas = 1
x = 0
for i in range (d+1):
    while x**2 <= n:
        x=x+pas
    x=x-pas
    print("Entre", round(x, i),"et", round(x+pas, i))
    pas=pas/10
```

2. Donner un encadrement de $\sqrt{2}$ et de $\sqrt{3}$ avec une amplitude de 10^{-6}

Exercice 2 : Méthode de dichotomie



On rappelle la propriété suivante :

Propriété : On considère une fonction f continue strictement monotone sur l'intervalle $[a; b]$ avec $f(a)f(b) < 0$, alors l'équation $f(x) = 0$ admet une unique solution α sur l'intervalle $[a; b]$ et pour tout $c \in]a; b[$:

- si $f(c) = 0$ alors $\alpha = c$.
- si $f(a)f(c) < 0$ alors $\alpha \in]a; c[$.
- si $f(c)f(b) < 0$ alors $\alpha \in]c; b[$.

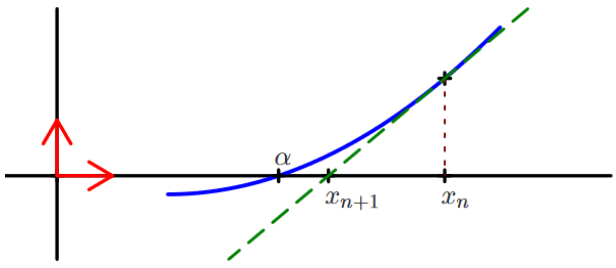
Afin d'encadrer la racine α de la fonction f sur l'intervalle $[a; b]$, on définit les suites $(a_n)_{n\in\mathbb{N}}$ et $(b_n)_{n\in\mathbb{N}}$:

$$\begin{cases} a_0 = a \\ b_0 = b \\ a_{n+1} = \begin{cases} a_n & \text{si } f(a_n)f\left(\frac{a_n+b_n}{2}\right) < 0 \\ \frac{a_n+b_n}{2} & \text{sinon} \end{cases} \\ b_{n+1} = \begin{cases} \frac{a_n+b_n}{2} & \text{si } f(a_n)f\left(\frac{a_n+b_n}{2}\right) < 0 \\ b_n & \text{sinon} \end{cases} \end{cases}$$

1. En quelle valeur s'annule la fonction $f(x) = x^2 - 2$? Donner un encadrement de cette valeur à l'unité près.
2. Programmer l'algorithme de dichotomie permettant d'obtenir un encadrement d'amplitude 10^{-6} de cette valeur.

Exercice 3 : Méthode de Newton

La *méthode de Newton* est une des méthodes algorithmiques de résolution d'équations. Elle vient palier au défaut majeur de la dichotomie, à savoir sa « lenteur ». Quel en est le principe ? Comment l'implémenter en Python ?



Considérons un point $A_n(x_n;f(x_n))$; l'équation de la tangente en ce point est:

$$y=f'(x_n)(x - x_n) + f(x_n)$$

et x_{n+1} est donc défini comme la solution de l'équation :

$$0=f'(x_n)(x_{n+1} - x_n) + f(x_n)$$

soit:

$$x_{n+1}=-\frac{f(x_n)}{f'(x_n)}x_n + x_n$$

Il faut donc, pour que cette méthode fonctionne, que tous les $f'(x_n)$ soient non nuls sur l'intervalle considéré.

Ainsi, nous allons considérer la suite x_n définie par:

$$\begin{cases} x_0 = a \\ x_{n+1} = -\frac{f(x_n)}{f'(x_n)} + x_n \end{cases}$$

Ecrire sous python une fonction Newton de paramètres f , Df , a et e calculant successivement les termes de la suite $(x_n)_{n\in\mathbb{N}}$ d'approximation d'un zéro de la fonction f de dérivée Df par la méthode de Newton en partant de $x_0 = a$ jusqu'à obtenir une différence des termes consécutifs inferieure à e en valeur absolue et retournant le dernier terme obtenu.

Donner une valeur approchée à 10^{-6} près de $\sqrt{2}$ à l'aide de cette fonction.