



Unité 3 : Débuter la programmation en Python

Compétence 1 : Fonctions et boucles

Dans cette première leçon de l'unité 3, vous allez mettre en œuvre vos connaissances en algorithmique et en langage Python afin de :

- Rechercher les solutions d'une équation $f(x) = 0$.
- Résoudre un problème d'optimisation.

Objectifs :

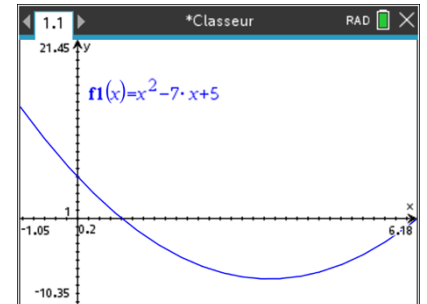
- Utiliser une fonction en langage Python.
- Mettre en œuvre la boucle bornée **While**.

Principe de la dichotomie

On considère la fonction f définie sur l'intervalle $[-2,3]$ par $f(x) = x^2 - 7x + 5$.

On utilise la calculatrice afin de tracer la courbe C_f représentant les variations de la fonction f .

Vous allez résoudre l'équation $f(x) = 0$ en écrivant un script Python correspondant à un algorithme connu et appelé « algorithme de dichotomie ».



Pour comprendre ce qu'est la dichotomie, on propose une petite expérience : « *chercher un mot dans un gros dictionnaire papier de 1024 pages* »

- Vous l'ouvrez au milieu : le mot ne s'y trouve pas, mais il est avant (il est donc dans les 512 premières pages).
- Vous ouvrez la moitié de la 1ère moitié : le mot ne s'y trouve pas, mais il est après (il est donc entre les pages 257 et 512).
- Vous ouvrez la moitié de la 2ème moitié... etc.

A chaque fois que vous progressez, le nombre de pages qui reste à examiner est divisé par 2.

Ainsi, dans un dictionnaire de 1024 pages, vous êtes certain de trouver votre page en 10 recherches seulement, puisque $1024/(2^{10})=1$





Algorithme :

Tant que $b - a > prec$ faire :

$$m \leftarrow \frac{a+b}{2}$$

Si $f(m)$ et $f(a)$ sont de signes, opposés

$$b \leftarrow m$$

sinon

$$a \leftarrow m$$

Fin Tant que

Commentaires :

$[a,b]$: Bornes de l'intervalle d'étude

f : Fonction étudiée, continue sur $[a ; b]$.

On se place au milieu de l'intervalle $[a ; b]$.

Si $f(m)$ et $f(a)$ sont de même signe, alors la solution de l'équation $f(x) = 0$ est située dans l'intervalle $[m ; b]$.

On se place donc sur $[m ; b]$

sinon.

on se place sur $[a ; m]$.

Mise en œuvre de l'algorithme :

- Encadrer entre deux entiers la valeur x_0 solution de l'équation $f(x) = 0$ avec une précision donnée « $prec$ ».
- Vous remarquerez que $f(0) \times f(1) < 0$
- Calculer $f(0) \times f\left(\frac{1}{2}\right)$ et $f\left(\frac{1}{2}\right) \times f(1)$
- En déduire si x_0 appartient à l'intervalle $\left[0 ; \frac{1}{2}\right]$ ou $\left[\frac{1}{2} ; 1\right]$

Créer un nouveau script et le nommer DICHO

- Entrer les différentes instructions, celles-ci se trouvent pour leur ensemble sous l'onglet (puis **4 Intégrés**).
- Les tests peuvent être obtenus directement par l'appui sur les touches puis **4 Intégrés** et enfin **3 Ops**.
- $[a, b]$ représente l'intervalle d'étude et n le nombre d'étapes.

```

1.1 1.2 *Classeur RAD
Dicho2.py enregistré avec succès
def f(x):
    return x**2-7*x+5
def dichot2(a,b,prec):
    while (b-a)>prec:
        c=(a+b)/2
        if f(a)*f(c)<=0:
            b=c
        else:
            a=c
    return a,b

```

Exécuter le script :

A partir de la représentation graphique de la fonction, tester le script.

```

1.1 1.2 1.3 *Classeur RAD
Shell Python 4/5
>>>#Running Dicho2.py
>>>from Dicho2 import *
>>>dichot2(0,1,0.01)
(0.8046875, 0.8125)
>>>

```





10 Minutes de Code

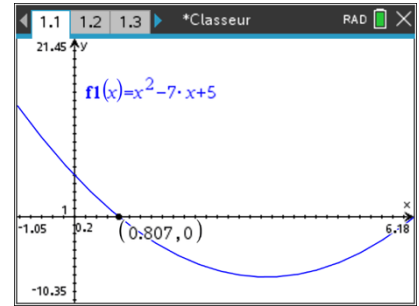
TI - NSPIRE™ CX II & TI - PYTHON

UNITE 3 : COMPETENCE 1

NOTES DU PROFESSEUR

Affiner votre recherche afin de trouver une solution proche de celle qui est proposée par la calculatrice.

La valeur exacte de x_0 dans l'intervalle $[0 ; 1]$ étant donnée par : $x_0 = \frac{7-\sqrt{29}}{2}$



Prolongements possibles :

- a) Au lieu de donner la précision, on peut travailler avec le nombre d'étapes.

```

1.2 1.3 1.4 *Classeur RAD
Dicho1.py enregistré avec succès
def f(x):
    return x**2-7*x+5
def dico1(a,b,n):
    for i in range(n):
        c=(a+b)/2
        if f(a)*f(c)<=0:
            b=c
        else:
            a=c
    return a,b

```

```

1.3 1.4 1.5 *Classeur RAD
Shell Python 2/5
>>>#Running Dicho1.py
>>>from Dicho1 import *
>>>dico1(0,1,25)
(0.8074175715446472, 0.8074176013469696)
>>>

```

La suite du script reste identique

- b) Envisager une programmation récursive

Remarque : Pour plus d'information sur la récursivité voir Compétence 3 de l'unité 3

```

def dico(a,b,prec) :
    if (b-a)<=prec :
        return a,b
    else :
        c=(a+b)/2
        if f(a)*f(b)<=0 :
            return dico(a,c,prec)
        else :
            return dico(c,b,prec)

```

