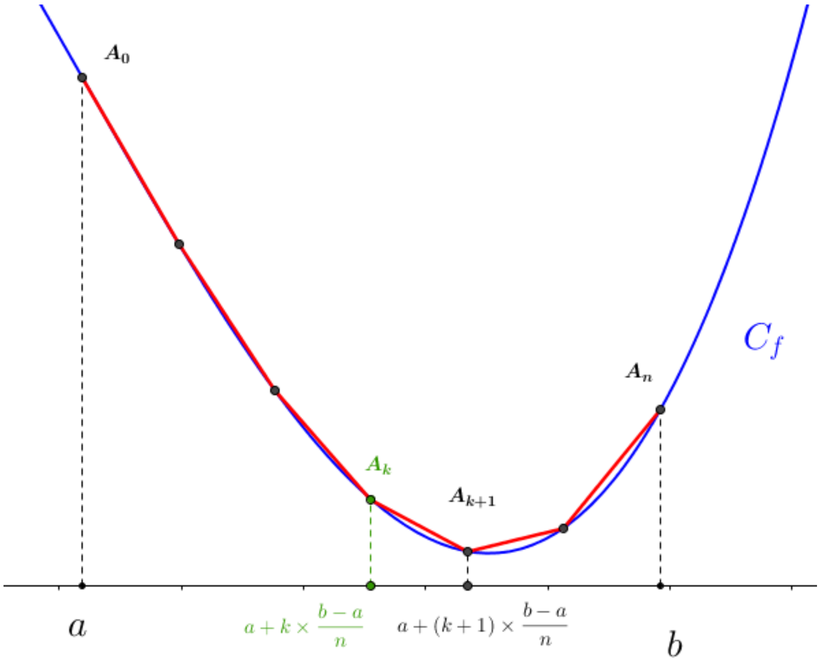
Calcul de la longueur d'une courbe.

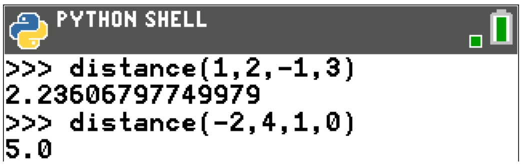
En se plaçant dans tout l'exercice dans un repère orthonormé $(O,\vec{\imath},\vec{\jmath})$, on va approcher la longueur d'une courbe d'un arc rectifiable à l'aide d'une ligne polygonale dont les extrémités sont des points de la courbe.



Dans un script LCOURBE

1°) Ecrire une fonction `d` qui prend comme paramètres les réels x_A, y_A, x_B, y_B et qui renvoie la distance AB.

Application : Calculer AB et CD avec $A\left|\begin{matrix}1\\2\end{matrix}\right.$, $B\left|\begin{matrix}-1\\3\end{matrix}\right.$, $C\left|\begin{matrix}-2\\4\end{matrix}\right.$ et $D\left|\begin{matrix}1\\0\end{matrix}\right.$



2°) Ecrire une fonction `f` qui prend comme paramètre le réel `x` et qui renvoie x^2+2x+3 .

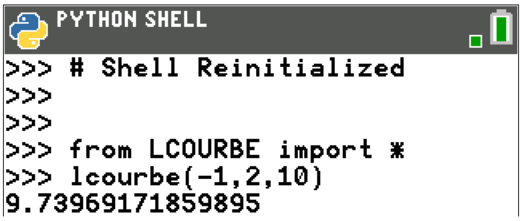
Application : Calculer $f(-1)$ et $f(2)$.

3°) Ecrire une fonction `lcourbe` qui prend comme paramètres les réels a, b et n un entier non nul et qui renvoie le calcul approché de la longueur de la courbe représentant la fonction `f` sur $[a,b]$ en partageant $[a,b]$ en n parties égales.

On rappelle :

$$l_{C_f}=\sum_{k=0}^{n-1}A_kA_{k+1}\quad\text{avec}\quad A_k\left|\begin{matrix}x_k\\f(x_k)\end{matrix}\right.\ \text{et}\ x_k=a+\frac{k(b-a)}{n}$$

Application avec $f(x)=x^2+2x+3$, $x\in[-1{,}2]$ et $n=10$ puis $n=100$ et $n=1000$



Thème: Analyse.

TI-83 Premium CE

Calcul de la longueur d'une courbe.

python

Fonction `d`

1°) On va utiliser la fonction racine carrée, il faut donc lire la bibliothèque math auparavant.

On rappelle que puissance 2 s'écrit `**2`.

```
ÉDITEUR : LCOURBE
LIGNE DU SCRIPT 0008
from math import *
def distance(xa,ya,xb,yb):
  d=sqrt((xb-xa)**2+(yb-ya)**2)
  return d
```

Fonction `f`

2°) On pouvait gagner une ligne de code en ne définissant pas `y` et en écrivant directement return `x**2+2*x+3`.

```
ÉDITEUR : LCOURBE
LIGNE DU SCRIPT 0015

def f(x):
  y=x**2+2*x+3
  return y
```

Fonction `lcourbe`

3°) On souhaite calculer :

$$\sum_{k=0}^{n-1} A_k A_{k+1} \text{ avec } A_k \left|_{f(x_k)}^{x_k}\right. \text{ et } x_k = a + \frac{k(b-a)}{n}$$

Il s'agit de calculer une somme qu'on va stocker dans la variable `s`.

On initialise `s` à 0.

Il y a `n` termes dans cette somme, on va donc utiliser une boucle `for i in range(n)` qui va tourner n fois.

A chaque tour de boucle il faut :

- Calculer x_k et x_{k+1}. On a nommé ces deux variables `x1` et `x2` dans le script.
- Calculer leurs images par f. On appelle `y1` et `y2` ces images.
- Ajouter à `s` la distance entre les deux points de coordonnées (`x1,y1`) et (`x2,y2`).

On renvoie la somme `s` à la fin du script.

On vérifie notre calcul en définissant la fonction dans l'application fonction de la TI-83 Premium CE :

Et à l'aide d'un calcul d'intégrale classique on vérifie notre résultat :

```
ÉDITEUR : LCOURBE
LIGNE DU SCRIPT 0019

def lcourbe(a,b,n):
  s=0
  for k in range(n):
    x1=a+k*(b-a)/n
    x2=a+(k+1)*(b-a)/n
    y1=f(x1)
    y2=f(x2)
    s=s+distance(x1,y1,x2,y2)
  return s
Fns...|a A #|Outils|Exéc|Script
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>>
>>>
>>> from LCOURBE import *
>>> lcourbe(-1,2,10)
9.73969171859895
>>> lcourbe(-1,2,100)
9.747014779047248
>>> lcourbe(-1,2,10000)
9.747088751210628
>>> |
Fns...|a A #|Outils|Éditer|Script
```

```
NORMAL FLOTT AUTO RÉEL RAD MP

 Graph1  Graph2  Graph3
■\Y1=X²+2X+3
```

```
NORMAL FLOTT AUTO RÉEL RAD MP
```

$$\int_{-1}^{2} \left(\sqrt{1+\left(\frac{d}{dX}(Y_1(X))\Big|_{X=X}\right)^2} \right) dX$$

9.74708876