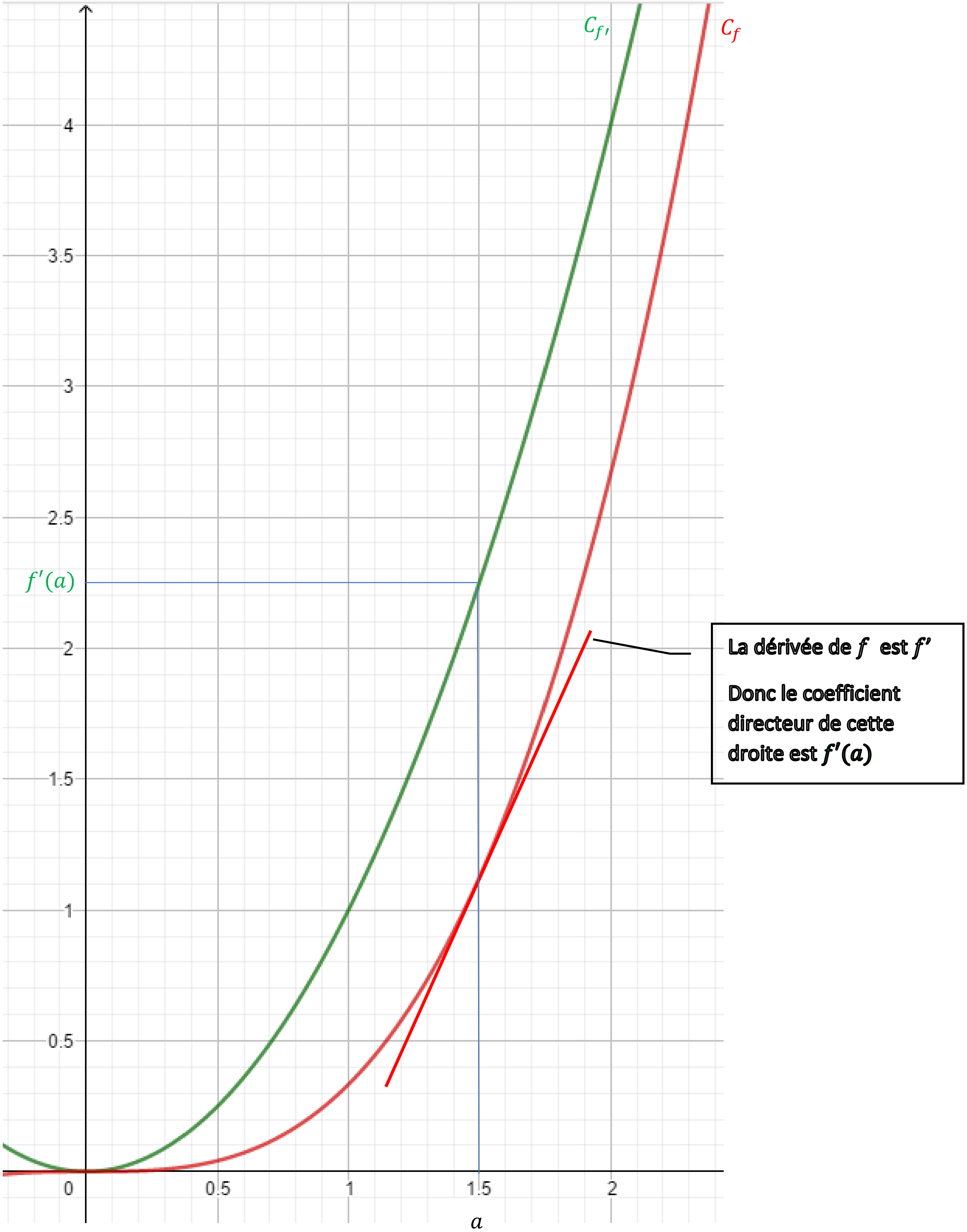
Corrigé

**Méthode d'Euler pour tracer la courbe de $f$ connaissant uniquement l'expression de $f'(x)$**

***Exemple :*** Cas déjà résolu : Soit $f'(x) = x^2$.

On connait déjà $f(x) = \frac{1}{3}x^3$ ce qui va nous servir pour contrôler notre approximation. Comment pourrait-on tracer une approximation de la courbe $C_f$ si on ne connaissait pas l'expression de $f(x)$, en sachant seulement que $f(0) = 0$ ?



*Réponse :*

On utilise un algorithme qui trace point par point une approximation de la courbe $C_f$.

Principe de l'algorithme :

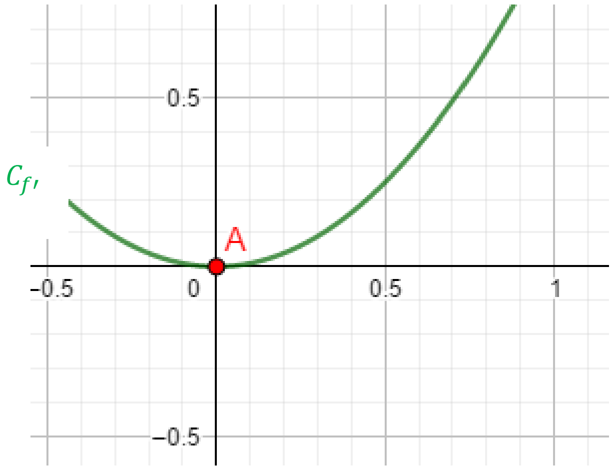
On se fixe « un pas » c'est-à-dire un réel strictement positif $h$ assez proche de zéro.

# 1 Premier exemple avec peu de points

On choisit $h = 0{,}5$

## 1.1 Tracé à la main de l'approximation de la courbe de la fonction $f$

On sait que $f(0) = 0$ donc la courbe $C_f$ passe par $A(0\,;0)$

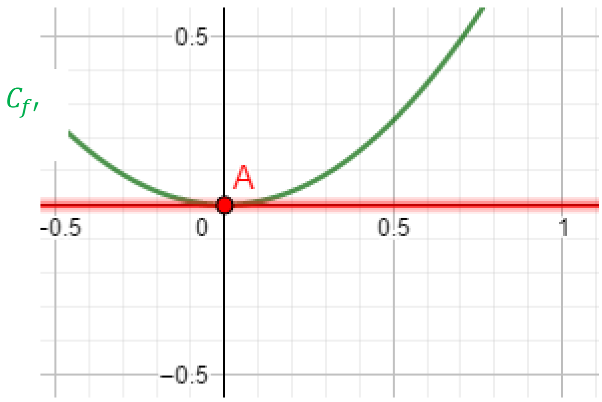



- On sait que $f'(0)$ est le nombre dérivé de $f$ calculé en $x = 0$.

On a $f'(0) = 0^2 = 0$.

Ce nombre dérivé est le coefficient directeur de la tangente à la courbe $C_f$ au point d'abscisse $0$.

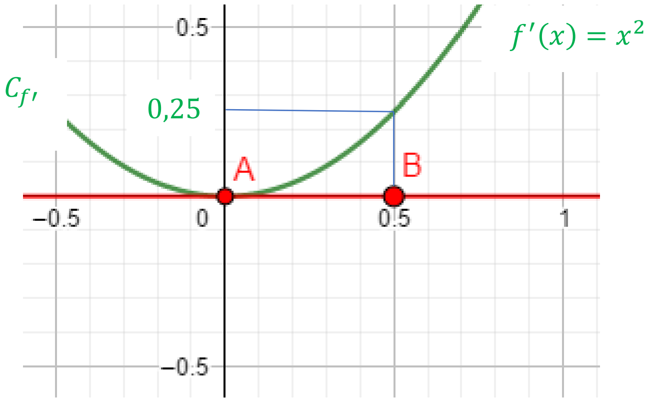
On trace cette tangente de coefficient directeur 0.




La tangente est a*pproximativement confondue* avec la courbe $C_f$

Donc provisoirement la tangente nous sert à tracer l'approximation de la courbe $C_f$

Comme le pas choisi est $h = 0{,}5$, on trace la tangente jusqu'à son point $B$ d'abscisse $0{,}5$.

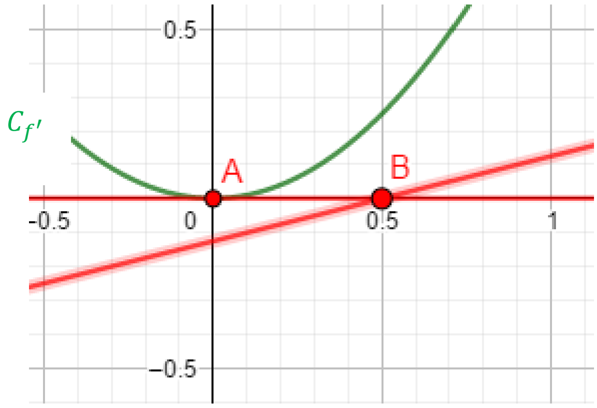



- $f'(0{,}5)$ est le nombre dérivé de $f$ calculé en $x = 0{,}5$

On a $f'(0{,}5) = 0{,}5^2 = 0{,}25$

Ce nombre dérivé nous donne la direction vers laquelle tracer notre approximation de courbe $C_f$.

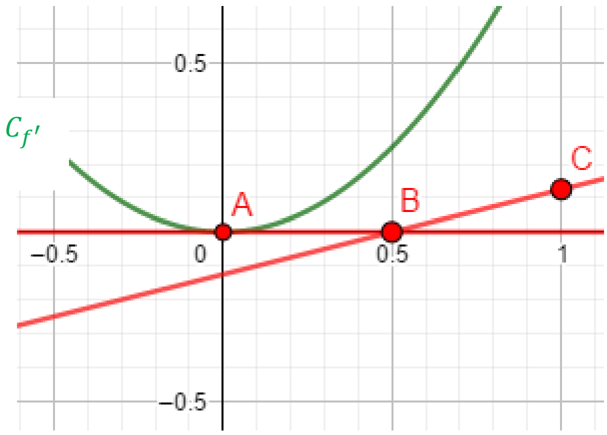
On trace la droite passant par $B$ ayant comme coefficient directeur $0{,}25$



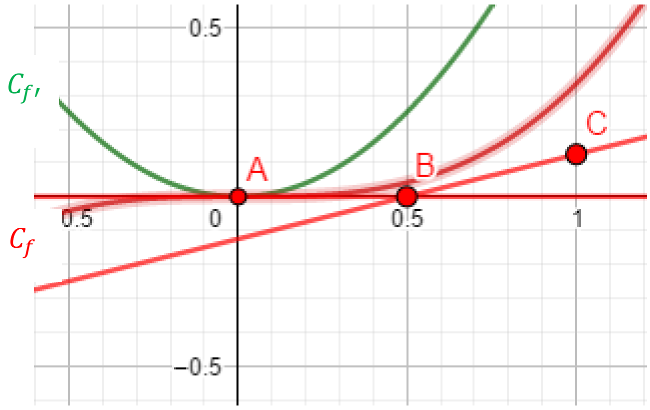

La droite est *approximativement confondue* avec la courbe $C_f$

Donc provisoirement la droite nous sert à tracer l'approximation de la courbe $C_f$

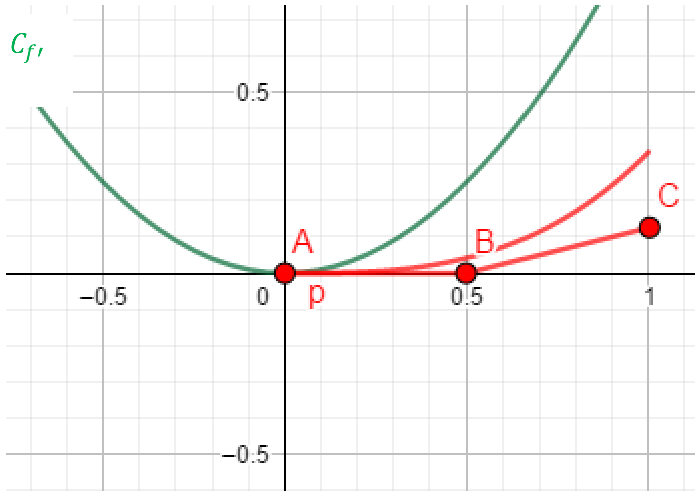
Comme le pas choisi est $h = 0{,}5$, on trace la droite jusqu'à son point $C$ d'abscisse $1$.



On peut comparer la ligne brisée ABC qui est *l'approximation de la courbe* $C_f$ avec la véritable courbe $C_f$ puisqu'ici nous connaissons l'expression de $f(x) = \frac{1}{3}x^3$




La différence entre la courbe $C_f$ et son approximation est assez importante. Cela est dû au fait que le pas choisi $h = 0{,}5$ est assez grand.

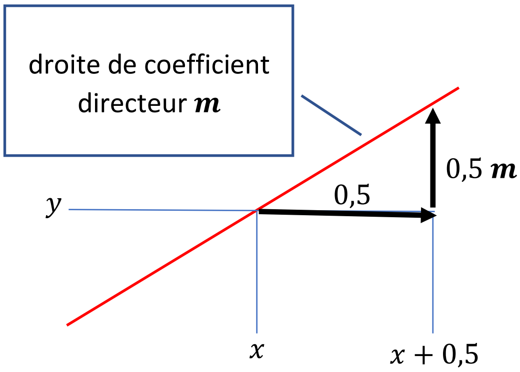



## 1.2 Algorithme pour calculer les coordonnées successivement des points $A, B, C$

```
def fprime(x):
    return x**2

def euler():
    y = 0
    x = 0
    print(x, y)
    for i in range(2):
        y = y + 0.5*fprime(x)
        x = x + 0.5
        print(x, y)
```

<u>Principe :</u>




écart en $y$ = écart en $x$ $\times$ **coefficient directeur**

On exécute la fonction `euler()`

Résultat :

`0    0`        (coordonnées de A)

`0.5    0.0`    (coordonnées de B)

`1.0    0.125`  (coordonnées de C)

## 2 Augmentation du nombre de points

**Exercice 1 Algorithme et tableur**

On modifie le programme précédent pour qu'il affiche les coordonnées des points de *l'approximation* de $C_f$ d'abscisses $0,\ 0.1,\ 0.2,\ 0.3, \dots ,\ 0.8,\ 0.9,\ 1$ (ce qui correspond à un pas $h = 0{,}1$).

1) Écrivez ci-dessous le programme modifié.

```python
def fprime(x):
    return x**2

def euler():
    y = 0
    x = 0
    print(x, y)
    for i in range(10):
        y = y + 0.1*fprime(x)
        x = x + 0.1
        print(x, y)
```

2) En faisant fonctionner le programme modifié, complétez le tableau ci-dessous :

| | A | B | C |
|---|---|---|---|
| 1 | n | xn | yn |
| 2 | 0 | 0 | 0 |
| 3 | 1 | 0.1 | 0 |
| 4 | 2 | 0.2 | 0.001 |
| 5 | 3 | 0.3 | 0.005 |
| 6 | 4 | 0.4 | 0.014 |
| 7 | 5 | 0.5 | 0.03 |
| 8 | 6 | 0.6 | 0.055 |
| 9 | 7 | 0.7 | 0.091 |
| 10 | 8 | 0.8 | 0.14 |
| 11 | 9 | 0.9 | 0.204 |
| 12 | 10 | 1 | 0.285 |

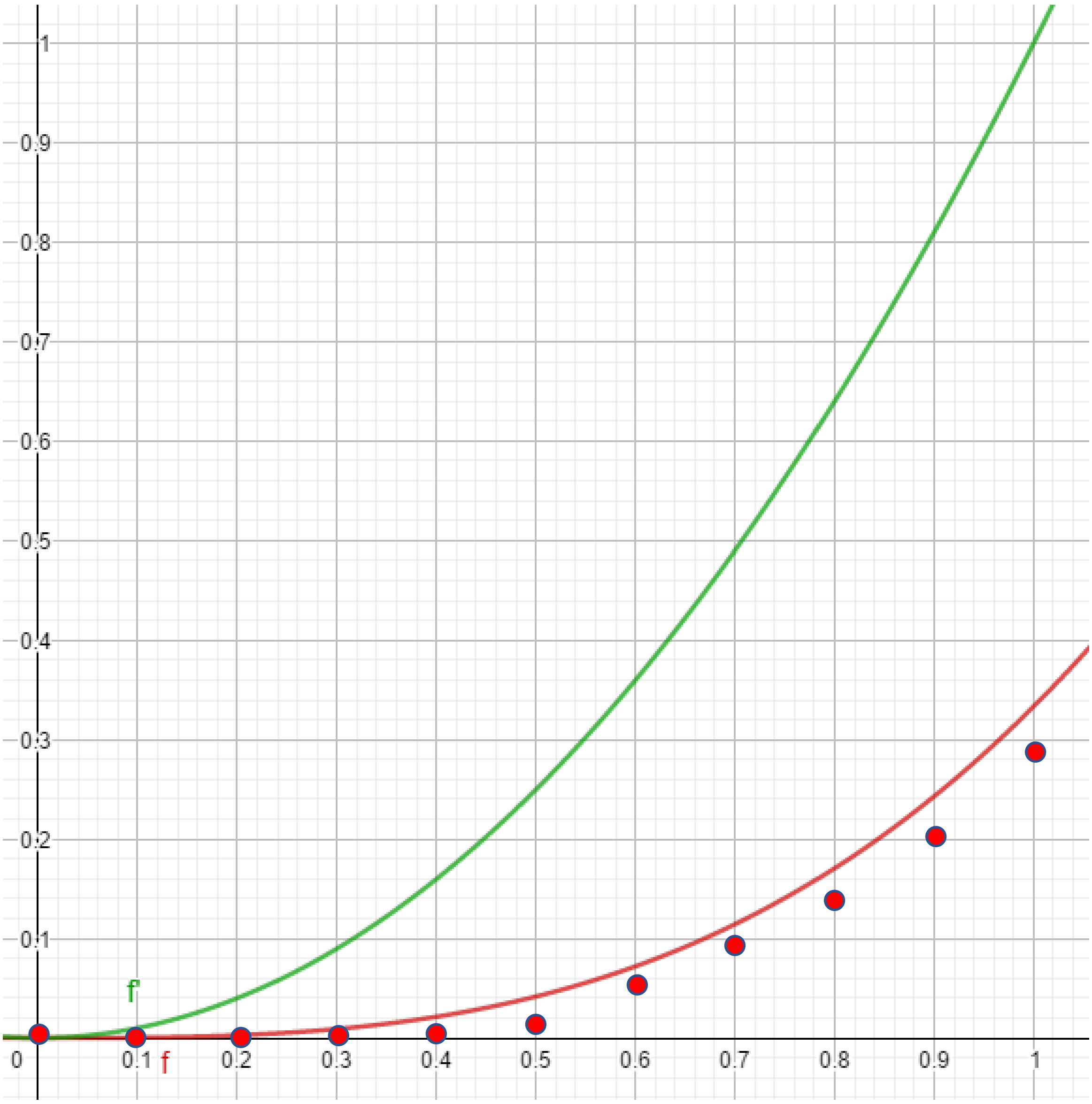
3) Si on voulait faire ces calculs dans une feuille de tableur, quelles formules devrait-on écrire dans les cellules B3 et C3 de manière à pouvoir les recopier vers le bas ?

En B3 on a la formule = B2 + 0.1 et en C3 on a la formule = C2 + 0.1 * B2^2

4) Donnez l'expression des suites $(x_n)$ et $(y_n)$ des coordonnées des points de la ligne brisée qui approche la courbe $C_f$

$$\begin{cases} y_0 = 0 \\ y_{n+1} = y_n + 0.1(x_n)^2 \end{cases} \quad \text{et} \quad \begin{cases} x_0 = 0 \\ x_{n+1} = x_n + 0{,}1 \end{cases}$$

5) Placez sur le graphique suivant les points correspondant aux coordonnées du tableau précédent.




6) Comparez la précision de l'approximation de $C_f$ obtenue avec le pas $h = 0{,}1$ et celle obtenue avec le pas $h = 0{,}5$.

Avec un pas de $h = 0{,}1$ l'approximation est beaucoup plus proche de la vraie courbe $C_f$

7) Que faudrait-il faire si on voulait que l'approximation se rapproche encore davantage de la véritable courbe $C_f$ ?

Il faudrait encore diminuer le pas $h$.

**On retient :**

| |
|---|
| ***Méthode d'Euler :*** $\boldsymbol{f(x_{n+1}) \approx f(x_n) + hf'(x_n)}$ |

**Exercice 2 Équation différentielle**

1) On considère la fonction $f$ qui est la solution de l'équation différentielle

$$y' = -0{,}1y$$

et qui vérifie la condition initiale $f(0) = 3$.

Dans cette question, on suppose qu'on ne sait pas trouver l'expression exacte de $f(x)$.

On va utiliser la méthode d'Euler pour tracer approximativement la courbe $C_f$ de la fonction $f$.

On a :

$$y_{n+1} \approx f(x_n) + hf'(x_n)$$

Exprimer $y_{n+1}$ uniquement en fonction de $h$ et de $f(x_n)$.

$y_{n+1} \approx f(x_n) + h \times -0{,}1f(x_n)$ $\quad\quad$ $y_{n+1} \approx f(x_n)(1 - 0{,}1h)$

2) Donnez en fonction de $h$ l'expression des suites $(x_n)$ et $(y_n)$ des coordonnées des points de la ligne brisée qui approche la courbe $C_f$.

$$\begin{cases} y_0 = 3 \\ y_{n+1} = y_n(1 - 0{,}1h) \end{cases} \quad \text{et} \quad \begin{cases} x_0 = 0 \\ x_{n+1} = x_n + h \end{cases}$$

3) On choisit le pas $h = 0{,}25$. Donnez l'expression des suites $(x_n)$ et $(y_n)$ des coordonnées des points de la ligne brisée qui approche la courbe $C_f$

$$\begin{cases} y_0 = 3 \\ y_{n+1} = 0{,}975\, y_n \end{cases} \quad \text{et} \quad \begin{cases} x_0 = 0 \\ x_{n+1} = x_n + 0{,}25 \end{cases}$$

4) On a utilisé un tableur pour calculer les coordonnées de la ligne brisée qui approche la courbe $C_f$ :

| | A | B | C |
|---|---|---|---|
| 1 | n | xn | yn |
| 2 | 0 | 0 | 3 |
| 3 | 1 | 0.25 | 2.925 |
| 4 | 2 | 0.5 | 2.851875 |

Quelles formules a-t-on écrit dans les cellules $B3$ et $C3$ de manière à pouvoir les recopier vers le bas ?

En B3 on a la formule = B2 + 0.25 et en C3 on a la formule = 0.975*C2

5) Quelle est la plus petite valeur de $x$ pour laquelle on a $f(x) < 1$ d'après ce tableau ?

On fait la table de la suite $(y_n)$ sur la calculatrice ; On trouve $n = 44$ ce qui donne $\boldsymbol{x = 11}$**.**

6) Résoudre de façon exacte l'équation différentielle

$$y' = -0{,}1y$$

et déterminer l'expression de la fonction $f$ qui vérifie la condition initiale $f(0) = 3$.

$f(x) = Ke^{-0{,}1x}$. $f(0) = 3$ équivaut à $Ke^{-0{,}1\times 0} = 3$ donc $K = 3$ d'où $f(x) = 3e^{-0{,}1x}$

7) Résoudre de façon exacte l'inéquation $f(x) < 1$.

$3e^{-0{,}1x} < 1$ donc $e^{-0{,}1x} < \frac{1}{3}$ donc $-0{,}1x < \ln\left(\frac{1}{3}\right)$ donc $-0{,}1x < -\ln(3)$ $x > 10\ln(3)$

8) La valeur de $x$ trouvée à la question 5 par la méthode d'Euler (qui est une méthode approximative) vous parait-elle bonne ?

$10\ln(3) \approx 10{,}986$ et l'approximation avait donné $x = 11$. Donc l'approximation parait bonne.

**Exercice 3 Équation différentielle dont *on ne connait pas* l'expression de la solution**

Pour construire une approximation de la courbe représentative d'une fonction $f$ en utilisant la méthode d'Euler, on construit une suite de points $M_n(x_n\,;\,y_n)$ tels que

$x_{n+1} = x_n + h$ avec $h$ réel positif et

$y_{n+1} = f(x_n + h)$ en utilisant l'approximation affine ***:*** $\boldsymbol{f(x_n + h) \approx f(x_n) + hf'(x_n)}$.

On considère l'équation différentielle :

$$(E) : y' = 4 - y^2$$

avec $y(0) = 0$.

- Ne connaissant pas la forme générale de la solution de cette équation, utilisez la méthode d'Euler[1] avec $h = 0{,}1$ pour programmer en Python le calcul des coordonnées permettant de tracer une approximation de la courbe représentative de cette solution sur l'intervalle[0 ; 2]. Vous écrirez votre programme ci-dessous.

On a :

$$y_{n+1} \approx f(x_n) + hf'(x_n)$$

On exprime $y_{n+1}$ uniquement en fonction de $h$ et de $f(x_n)$.

$$y_{n+1} \approx f(x_n) + h \times (4 - \left(f(x_n)\right)^2)$$

$h = 0{,}1$ donc $y_{n+1} \approx f(x_n) + 0{,}1 \times (4 - \left(f(x_n)\right)^2)$

$$y_{n+1} \approx f(x_n) + 0{,}4 - 0{,}1\left(f(x_n)\right)^2$$

$$\begin{cases} y_0 = 0 \\ y_{n+1} = y_n + 0{,}4 - 0{,}1y_n^2 \end{cases} \quad \text{et} \quad \begin{cases} x_0 = 0 \\ x_{n+1} = x_n + 0{,}1 \end{cases}$$

D'où le programme Python :

[1] Léonard Euler : Mathématicien suisse (1707 – 1783)

```python
def programme():
    y = 0
    x = 0
    print(x, y)
    for i in range(20):
        y = y + 0.4 - 0.1*(y**2)
        x = x + 0.1
        print(x, y)

programme()
```

Résultat :

0 0

0.1 0.4

0.2 0.784

0.30000000000000004 1.1225344000000002

0.4 1.396526052081664

0.5 1.601497550667384

0.6 1.745018110188021

0.7 1.840509289699604

0.7999999999999999 1.90176184515255

0.8999999999999999 1.940092033584747

0.9999999999999999 1.963696323706847

1.0999999999999999 1.9780859985328685

1.2 1.986803576773691

1.3 1.992064731505618

1.4000000000000001 1.995232542054763

1.5000000000000002 1.9971372523673316

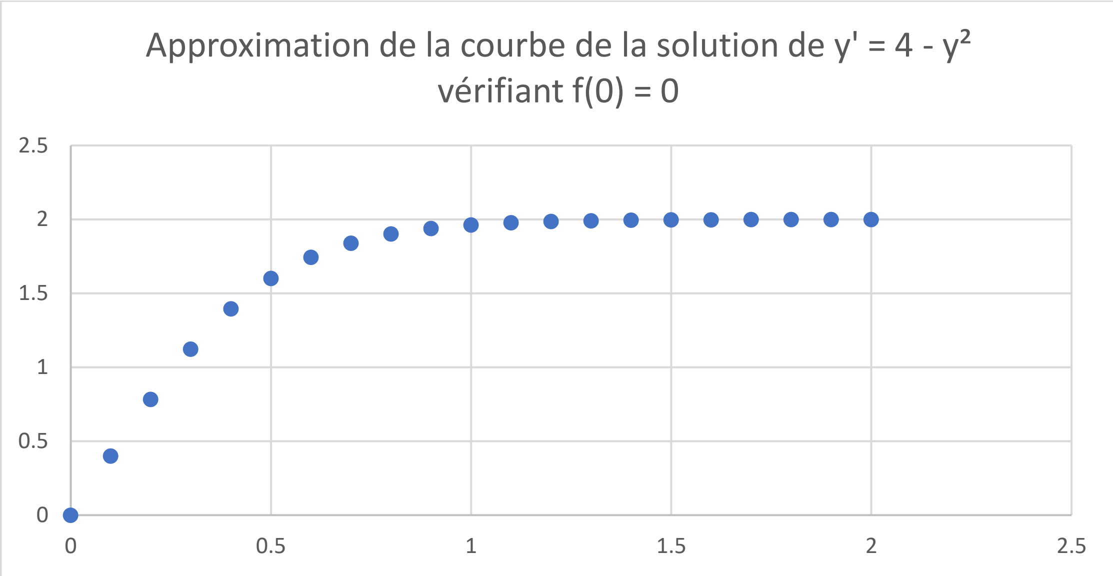
1.6000000000000003 1.9982815318879983

1.7000000000000004 1.9989686238195339

1.8000000000000005 1.9993810679180377

1.9000000000000006 1.9996286024431305

2.0000000000000004 1.9997771476722637




Sur la calculatrice TI 83  (système d'exploitation au moins version 5.5):

ÉDITEUR : PRIMITIV
LIGNE DU SCRIPT 0011

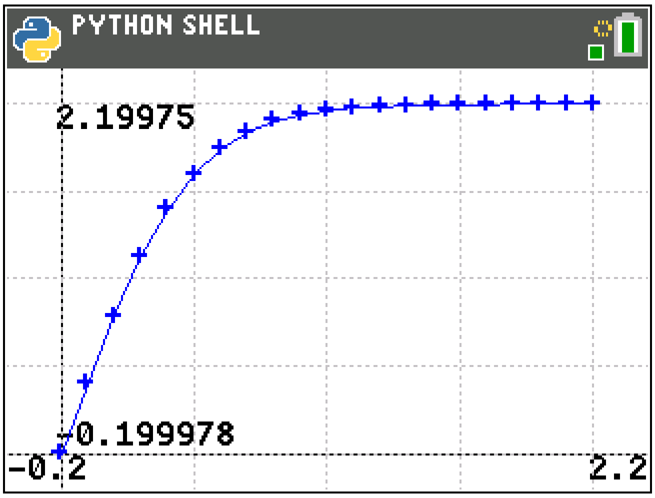
```
import ti_plotlib as plt

def eulersim():
  x=0
  y=0
  xs=[x]
  ys=[y]
  for i in range(20):
    y=y+0.4-0.1*(y**2)
    x=x+0.1
    xs.append(x)
    ys.append(y)
  return xs,ys

def principal():
  lx,ly=eulersim()
  plt.cls()
  plt.auto_window(lx,ly)
  plt.axes("on")
  plt.grid(0.5,0.5,"dot")
  plt.color(0,0,255)
  plt.plot(lx,ly,"+")
  plt.show_plot()
```

| Fns… | a A # | Outils | Exéc | Script |
|---|---|---|---|---|

Après exécution, on obtient :


PYTHON SHELL
2.19975
-0.199978
-0.2
2.2