

LES SUITES NUMÉRIQUES M06

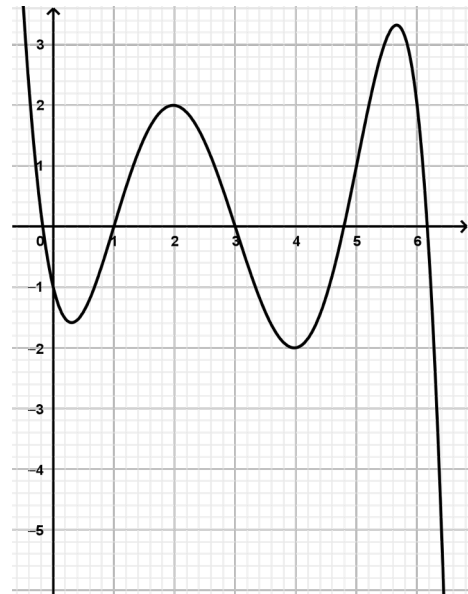
EXERCICE N°1 Lecture graphique

[VOIR LE CORRIGÉ](#)

On a représenté ci-contre une fonction f .

On définit une suite u par : $\forall n \in \mathbb{N}, u_n = f(n)$

On admet que $u_0 = -1$.



EXERCICE N°2 Utiliser un graphique (méthode à connaître)

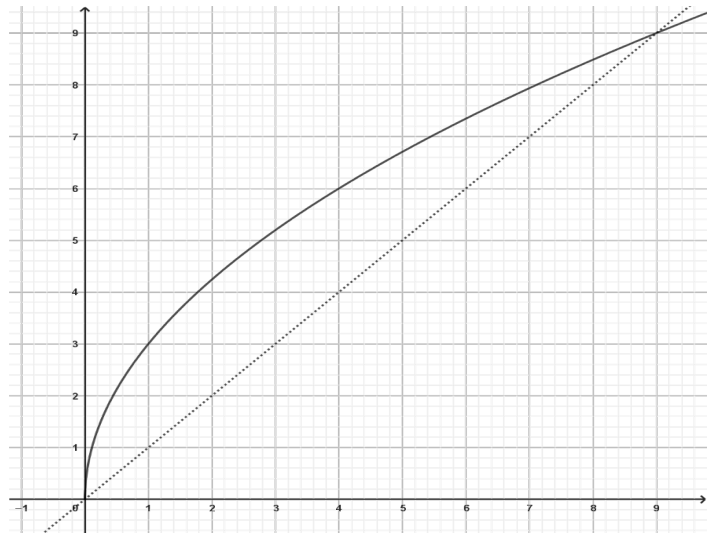
[VOIR LE CORRIGÉ](#)

(On a représenté une fonction g ainsi que la droite d'équation $y = x$ dans le graphique ci-contre.

On définit la suite v par :

$$\begin{cases} v_0 = 2 \\ \forall n \in \mathbb{N}, v_{n+1} = g(v_n) \end{cases}$$

Déterminer les cinq premiers termes de la suite v .



(version imprimable en cliquant sur le graphique)

EXERCICE N°3 Un peu de python

[VOIR LE CORRIGÉ](#)

Soit $(u_n)_{n \in \mathbb{N}}$ définie par : $u_0 = 2$ et $u_{n+1} = 2u_n - n$.

On considère l'algorithme ci-contre :

- 1) Que permet d'afficher cet algorithme ?
- 2) Quelle valeur affiche cet algorithme ?
- 3) Coder cet algorithme en python et vérifier la question 2.
- 4) Modifier le script Python afin qu'il affiche uniquement la dernière valeur.
- 5) Modifier le code Python afin $u_0 = 3$ que vaut alors u_{10} ?

```
u ← 2
Pour i allant de 1 à 10
    u ← 2u - i
    Afficher u
Fin pour
```



basthon

EXERCICE N°4 Encore un peu de python

[VOIR LE CORRIGÉ](#)

Soit u la suite définie par $\begin{cases} u_0 = 5 \\ \forall n \in \mathbb{N}, u_{n+1} = 3u_n - 7 \end{cases}$

- 1) Calculer u_1 et u_2 .
- 2) Écrire un algorithme permettant de calculer u_{20} .
- 3) Coder cet algorithme en Python et l'utiliser pour calculer u_{20} .
- 4) Modifier le script Python afin qu'il renvoie la liste contenant les 21 premiers termes de u .



basthon

LES SUITES NUMÉRIQUES M06C

EXERCICE N°1 Lecture graphique

[RETOUR À L'EXERCICE](#)

On a représenté ci-contre une fonction f .

On définit une suite u par : $\forall n \in \mathbb{N}, u_n = f(n)$

On admet que $u_0 = -1$.

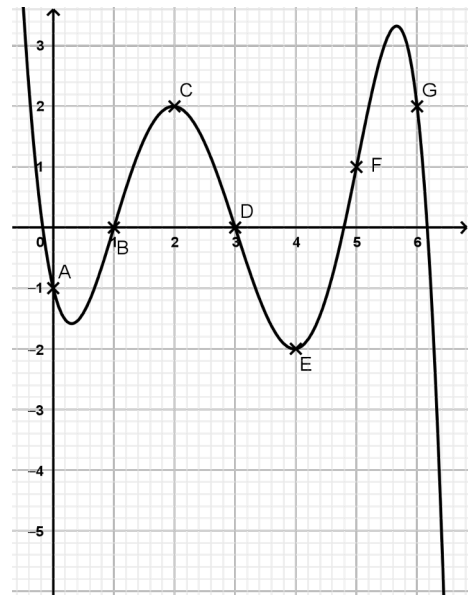
Donner les valeurs des six termes suivants.

Les termes de la suite sont représentés par les points.

$A(0, -1)$ correspond à $u_0 = -1$ et ainsi de suite.

$$u_1 = 0 ; u_2 = 2 ; u_3 = 0$$

$$u_4 = -2 ; u_5 = 1 \text{ et } u_6 = 2$$



LES SUITES NUMÉRIQUES M06C

EXERCICE N°2 Utiliser un graphique (méthode à connaître)

[RETOUR À L'EXERCICE](#)



scannez
ou
cliquez

(On a représenté une fonction g ainsi que la droite d'équation $y = x$ dans le graphique ci-contre.

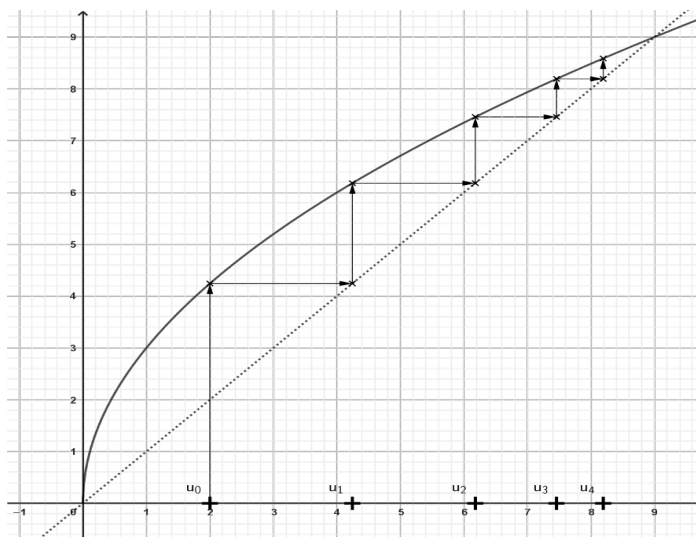
On définit la suite v par :

$$\begin{cases} v_0 = 2 \\ \forall n \in \mathbb{N}, v_{n+1} = g(v_n) \end{cases}$$

Déterminer les cinq premiers termes de la suite v .

$$u_0 = 2 ; u_1 \approx 4,2 ; u_2 \approx 6,2$$

$$u_3 \approx 7,5 \text{ et } u_5 \approx 8,2$$



(version imprimable en cliquant sur le graphique)

LES SUITES NUMÉRIQUES M06C

EXERCICE N°3 Un peu de python

[RETOUR À L'EXERCICE](#)



basthon

Soit $(u_n)_{n \in \mathbb{N}}$ définie par : $u_0 = 2$ et $u_{n+1} = 2u_n - n$.

On considère l'algorithme ci-contre :

1) Que permet d'afficher cet algorithme ?

Il permet d'afficher les valeurs de u_1 jusqu'à u_{10} .

2) Quelle valeur affiche cet algorithme ?

3, 4, 5, 6, 7, 8, 9, 10, 11, 12

3) Coder cet algorithme en python.

```
1 u = 2
2 for i in range(1,11): #Ne pas oublier la petite subtilité
3     u = 2*u-i
4     print(u)
```

4) Modifier le script Python afin qu'il affiche uniquement la dernière valeur.

```
1 u = 2
2 for i in range(1,11): #Ne pas oublier la petite subtilité
3     u = 2*u-i
4     print(u)
```

Voyez l'importance de l'indentation (ligne 4).

5) Modifier le code Python afin $u_0 = 3$ que vaut alors u_{10} ?

```
1 u = 3
2 for i in range(1,11): #Ne pas oublier la petite subtilité
3     u = 2*u-i
4     print(u)
```

$u_{10} = 1036$

```
u ← 2
Pour i allant de 1 à 10
    u ← 2u - i
    Afficher u
Fin pour
```

LES SUITES NUMÉRIQUES M06C

EXERCICE N°4 Encore un peu de python

[RETOUR À L'EXERCICE](#)



basthon

Soit u la suite définie par
$$\begin{cases} u_0 = 5 \\ \forall n \in \mathbb{N}, u_{n+1} = 3u_n - 7 \end{cases}$$

1) Calculer u_1 et u_2 .

- $u_1 = 2u_0 + 1 = 2 \times 5 + 1$, ainsi $u_1 = 11$
- $u_2 = 2u_1 + 1 = 2 \times 11 + 1$, ainsi $u_2 = 23$

2) Écrire un algorithme permettant de calculer u_{20} .

```
u ← 5
Pour i allant de 1 à 20
    u ← 3 × u - 7
Fin pour
Afficher u
```

3) Coder cet algorithme en Python et l'utiliser pour calculer u_{20}

```
1 u = 5
2 for i in range(1,21): #Ne pas oublier la petite subtilité
3     u = 3*u-7
4 print(u)
```

4) Modifier le script Python afin qu'il renvoie la liste contenant les 21 premiers termes de u .

```
1 u = [5]
2 for i in range(1,21): #Ne pas oublier la petite subtilité
3     u.append(3*u[-1]-7)
4 print(u)
```

ANNEXE À IMPRIMER

