Fiche S1.1 : La notion de suite

1 – Définition d'une suite

Rappel : On note $\mathbb{N}$ l'ensemble des entiers naturels, c'est-à-dire $\mathbb{N} = \{0; 1; 2; 3; 4; 5; \dots\}$

Définition 1 : Une **suite numérique**, notée (u_n), est une fonction définie sur $\mathbb{N}$ (ou une partie de $\mathbb{N}$). L'image d'un entier naturel n est notée u_n et est appelée le **terme de rang** n.

Remarques :

● Une suite est une fonction dont la **variable** n est un entier positif ou nul.

● On peut voir une suite comme une **liste indexée** de nombres réels :

$$(u_n) = (\underbrace{u_0}_{\substack{1\text{er} \\ \text{terme}}} ; u_1 ; u_2 ; \dots ; \underbrace{u_5}_{\substack{\text{terme de} \\ \text{rang } 5}} ; \dots ; \underbrace{u_{n-1}}_{\substack{\text{terme} \\ \text{précédent}}} ; \underbrace{u_n}_{\substack{\text{terme de} \\ \text{rang } n}} ; \underbrace{u_{n+1}}_{\substack{\text{terme} \\ \text{suivant}}} ; \dots)$$

Exemple 1 : Soit (d_n) la suite des décimales du nombre π.

On peut écrire $(d_n) = (3 ; 1 ; 4 ; 1 ; 5 ; 9 ; 2 ; \dots)$. Les premiers termes sont $d_0 = 3$, $d_1 = 1$, $d_2 = 4$, etc.

Mode de génération : On peut définir une suite de deux façons différentes :

● Par une **formule explicite** : $u_n = f(n)$.

● Par une **relation de récurrence**, à l'aide des deux données suivantes :

◦ Le premier terme u_0.

◦ Une relation qui définit chaque terme en fonction des précédents : $u_{n+1} = f(u_n)$.

Exemple 2 : Soit (u_n) la suite définie pour tout entier naturel n par $u_n = n^2 + 2n - 5$

C'est une relation explicite : On a $u_n = f(n)$ avec $f(x) = x^2 + 2x - 5$.

Calculons les premiers termes de la suite :

$u_0 = f(0) = 0^2 + 2 \times 0 - 5 = -5$

$u_1 = f(1) = 1^2 + 2 \times 1 - 5 = -2$

$u_2 = f(2) = 2^2 + 2 \times 2 - 5 = 3$

$u_3 = f(3) = 3^2 + 2 \times 3 - 5 = 10$

Remarque : Pour calculer un terme de la suite, il suffit de remplacer n par le rang souhaité dans la formule.

Exemple 3 : Soit (v_n) la suite définie pour tout entier naturel n par $\begin{cases} v_0 = 1 \\ v_{n+1} = 2v_n + 1 \end{cases}$

La suite (v_n) est définie par une relation de récurrence.

On donne le premier terme puis chaque terme s'obtient à partir du précédent.

Calculons les premiers termes de la suite :

$v_1 = 2 \times v_0 + 1 = 2 \times 1 + 1 = 3$

$v_2 = 2 \times v_1 + 1 = 2 \times 3 + 1 = 7$

$v_3 = 2 \times v_2 + 1 = 2 \times 7 + 1 = 15$

Remarque : Pour calculer un terme de la suite, il faut calculer tous les termes précédents.

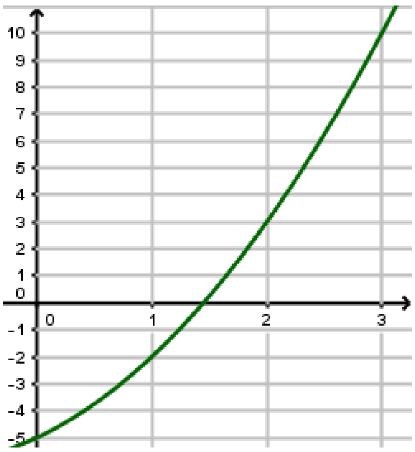
2 – Représentation graphique d'une suite

Définition 2 : Dans un repère du plan, la représentation graphique d'une suite (u_n) est l'ensemble des points qui sont de coordonnées $(n; u_n)$.

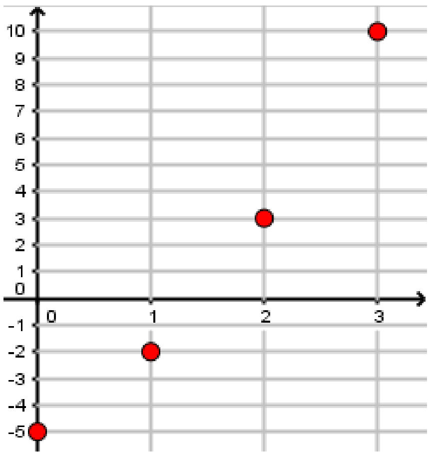
Remarque : La représentation graphique d'une fonction est une courbe alors que la représentation graphique d'une suite est un **nuage de points.**

Exemple 4 : Représentation graphique de la suite (u_n)

Fonction $f(x) = x^2 + 2x - 5$



Suite $u_n = n^2 + 2n - 5$



3 – Calcul des termes à l'aide d'un algorithme

Il est possible de générer les termes d'une suite à l'aide d'un algorithme

Exemple 5 : Algorithme qui calcule et affiche les 100 premiers termes de la suite explicite (u_n).

Algorithme

Variables : $\overset{rang}{\overbrace{n}}$, $\overset{terme}{\overbrace{u}}$
Pour n allant de 0 *à* 99
  $u \longleftarrow n^2 + 2n - 5$
  Afficher u
Fin Pour

Exécution

● 1er tour de boucle : $n \leftarrow 0$
$u \leftarrow 0^2 + 2 \times 0 - 5 = -5$
« -5 » (On affiche u_0)

● 2eme tour de boucle : $n \leftarrow 1$
$u \leftarrow 1^2 + 2 \times 1 - 5 = -2$
« -2 » (On affiche u_1) etc.

Programme Python

```
from math import*

for n in range(0,100):
  u=n**2+2*n-5
  print(u)

# Attention pour aller de 0 à 99
# On doit écrire range(0,100)
```

Exemple 6 : Algorithme qui calcule et affiche les 100 premiers termes de la suite récurrente (v_n).

Algorithme

Variables : $\overset{rang}{\overbrace{n}}$, $\overset{terme}{\overbrace{v}}$
$v \leftarrow 1$
Afficher v
Pour n allant de 1 *à* 99
  $v \longleftarrow 2v + 1$
  Afficher v
Fin Pour

Exécution

$v \leftarrow 1$
« 1 » (On affiche v_0)

● 1er tour de boucle : $n \leftarrow 0$
$v \leftarrow 2 \times 1 + 1 = 3$
« 3 » (On affiche v_1)

● 2eme tour de boucle : $n \leftarrow 1$
$v \leftarrow 2 \times 3 + 1 = 7$
« 7 » (On affiche v_2) etc.

Programme Python

```
from math import*

v=1
print(v)
for n in range(1,100):
  v=2*v+1
  print(v)

# Attention pour aller de 1 à 99
# On doit écrire range(1,100)
```