Introduction aux suites numériques, cours, classe de première spécialité Mathématiques

F.Gaudon

30 juin 2019

Table des matières

1 Notion de suite **2**

2 Méthodes de construction des suites **2**

2.1 Définition explicite 2

2.2 Définition par récurrence 4

3 Représentation graphique **6**

1 Notion de suite

Définition :
On appelle *suite* toute application u de $\mathbb{N}$ dans $\mathbb{R}$.

Exemple :

$$\begin{aligned} u : \mathbb{N} &\longrightarrow \mathbb{R} \\ n &\longmapsto 3n^2+4 \end{aligned}$$

On a $u_0 = 3\times 0^2+4 = 4$, $u_1 = 3\times 1^2+4 = 7$, $u_5 = 3\times 5^2+4 = 79$.

Définition :

- L'image de n par la suite u est notée u_n ou $u(n)$.
- u_n est appelé *terme* de la suite
- La suite u est notée $(u_n)_n \in \mathbb{N}$ ou $(u_n)_n$.

Remarque :

- Si u_0 est le premier *terme* de la suite, u_n est le $n+1^{\text{e}}$ terme.
- Si u_1 est le premier *terme* de la suite, u_n est le n^{e} terme.

2 Méthodes de construction des suites

2.1 Définition explicite

Définition :
Soit f une fonction de $\mathbb{R}^+$ dans $\mathbb{R}$, on définit une suite $(u_n)_n$ en posant pour tout $n \in \mathbb{N}$, $u_n = f(n)$.

Exemple [Calculer des termes d'une suite définie explicitement] :

$$\begin{aligned} u : \mathbb{N} &\longrightarrow \mathbb{R} \\ n &\longmapsto u_n = 3n \end{aligned}$$

On a $u_0 = 3\times 0-2 = -2$
$u_1 = 3\times 1-2 = 1$
$u_6 = 3\times 6-2 = 16$.

Algorithme de calcul de terme :
Algorithme d'obtention du terme de rang n d'une suite (u_n) définie à partir d'un rang p et définie par $u_n = f(n)$ pour tout $n \geq p$.

Données : n, f, p
Début traitement
| Si $n \geq p$ alors $u \leftarrow f(n)$;
Fin

Exemple de programmation en langage python :
Soit (u_n) définie par $u_n = 3n + 2$ pour tout entier naturel n. p désigne le premier rang de la suite,0 ici. n désigne le rang dont on cherche à calculer le terme et u désigne le terme de la suite.

```
def calculTerme(p,n):
        if n>=p:
                u=3*n+2
                return u
```

Algorithme de calcul d'une liste de termes :
Algorithme d'obtention de la liste des termes jusqu'à un rang n donné pour une suite (u_n) définie explicitement par $u_n = f(n)$ et son premier terme u_p.

Données : p, n, f
Début traitement
| L est une liste vide ;
| **pour** k ***allant de*** p ***à*** n **faire**
| | $u \leftarrow f(k)$;
| | Ajouter u à la liste L ;
| **fin**
Fin

Exemple de programmation en python :
Soit (u_n) définie par $u_{n+1} = 3n + 2$ pour tout entier naturel n. p désigne le premier rang de la suite, 0 ici. n désigne le dernier rang dont on cherche à obtenir le terme et u désigne les différents termes de la suite.

- Construction en extension, avec une boucle bornée :

```
def obtientionListe(p,n):
        L=[]
        for k in range(p,n+1):
                u=3*k+2
                L.append(u)
        return L
```

- Construction en extension, avec une boucle non bornée :

```
def obtentionListe(p,n):
        L=[]
        k=p
        while k<n:
                u=3*k+2
                L.append(u)
                k=k+1
        return L
```

- Construction en compréhension :

```
def obtentionListe(p,n):
        L=[3*k+2 for k in range(p,n+1)]
        return L
```

2.2 Définition par récurrence

Définition :
Soit f une fonction de $\mathbb{R}$ dans $\mathbb{R}$. Une suite définie par *récurrence* est une suite définie par la donnée de son premier terme u_p où p est un entier naturel et par la relation pour tout n entier naturel tel que $n \geq p$, $u_{n+1} = f(u_n)$.

Exemple [Calculer les premiers termes d'une suite définie par récurrence] :

$$u_n = \begin{cases} u_0 = 4 \\ u_{n+1} = 3u_n - 2 \end{cases}$$

On a $u_1 = 3 \times 4 - 2 = 10$
$u_2 = 3 \times 10 - 2 = 28$
$u_3 = 3 \times 28 - 2 = 82$.

Algorithme de calcul d'un terme de rang donné :
Algorithme d'obtention du terme de rang n d'une suite (u_n) définie à partir d'un rang p et définie par $u_{n+1} = f(u_n)$ pour tout $n \geq p$.

Données : p, n, u_p, f
Début traitement
  $u \leftarrow u_p$;
  pour k ***allant de*** $p+1$ ***à*** n **faire**
    $u \leftarrow f(u)$;
  fin
Fin

Exemple de programmation en langage python :

Soit (u_n) définie par $u_{n+1} = 3u_n + 2$ pour tout entier naturel n non nul et par $u_1 = 2$ n. p désigne le premier rang de la suite (1 ici), n désigne le rang dont on cherche à calculer le terme et u désigne les différents termes de la suite.

- Avec une boucle bornée :

```
def calculTerme(u,p,n):
        for k in range(p,n):
                u=3*u+2
        return u
```

- Avec une boucle non bornée :

```
def calculTerme(u,p,n):
        k=p
        while k<n:
                u=3*u+2
                k=k+1
        return u
```

Algorithme d'obtention de la liste des premiers termes :

Algorithme d'obtention de la liste des termes jusqu'à un rang n donné pour une suite (u_n) définie par récurrence par $u_{n+1} = f(u_n)$ et son premier terme u_p.

Données : p, n, u_p, f
Début traitement
  $u \leftarrow u_p$;
  L est une liste réduite à u ;
  pour k ***allant de*** $p+1$ ***à*** n **faire**
    $u \leftarrow f(u)$ Ajouter u à la liste L ;
    ;
  fin
Fin

Exemple de programmation en python :

Soit (u_n) définie par $u_{n+1} = 3u_n + 2$ pour tout entier naturel n non nul et par $u_1 = 2$ n. p désigne le premier rang de la suite (1 ici), n désigne le dernier rang dont on cherche à obtenir le terme et u désigne les différents termes de la suite. On ne peut utiliser que des constructions en extension ici.

- Avec une boucle bornée :

```
def obtentionListe(u,p,n):
        L=[u]
        for k in range(p,n):
                u=3*u+2
                L.append(u)
        return L
```

- Avec une boucle non bornée :

```
def obtentionListe(u,p,n):
        L=[u]
        k=p
        while k<n:
                u=3*u+2
                k=k+1
                L.append(u)
        return L
```

3 Représentation graphique

Définition :
La représentation graphique d'une suite (u_n) dans un repère est l'ensemble des points de coordonnées $(n; u_n)$.

Exemple :
La figure ci-dessous montre la représentation graphique de la suite définie par $u_n = -4 \times \frac{1}{2^n}$ pour tout entier naturel n.